

Creation of a Custom Language Model for Pediatric Occupational Therapy Documentation

by

Rachel DiMaio

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2024

© Rachel DiMaio 2024

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Rachel DiMaio was the sole author for all Chapters of this thesis which were written under the supervision of Dr. Bryan Tripp and were not written for publication.

This research was conducted at the University of Waterloo by Rachel DiMaio under the supervision of Dr. Bryan Tripp and in collaboration with the KidsAbility Research and Innovation Team supervised by Brendan Wylie-Toal. Dr. Bryan Tripp, Rachel DiMaio, Tia Tuinstra, Trevor Yu, and Brendan Wylie-Toal contributed to study design and planning throughout the research. Rachel DiMaio and Tia Tuinstra performed development work for the research, with Brendan Wylie-Toal and his Innovation Team facilitating the Pilot Study at the KidsAbility Center and contributing to data collection. The development work and data analysis presented in this thesis is the independent work of Rachel DiMaio with minor exceptions as follows:

- Trevor Yu and Tia Tuinstra contributed to the evaluation of some model outputs in Chapter 5, particularly for the comparison of the Llama 2 and 3 fine-tuned models.
- Tia Tuinstra contributed by developing the web application for the model that is mentioned in Chapter 6.
- A study conducted by Solomon Amenyo et al. from the University of Waterloo is referenced in Chapter 6 as the source of the data used for quality analysis of the model outputs.

Abstract

KidsAbility is a pediatric rehabilitation center that offers services including occupational therapy (OT) to youth. Documentation, including writing progress notes for each treatment appointment, is essential to OT treatment but can also be time-consuming and tedious. If the time spent on writing progress notes was reduced, KidsAbility believes that their capacity for treatment would increase. This thesis explores the creation of a custom large language model that is intended to decrease the amount of time that clinicians spend writing progress notes by transforming point-form scratch notes from pediatric OT treatment appointments into draft full-form documentation in SOAP format for the clinicians to edit.

A dataset of thousands of historical progress notes, with personal health information redacted, was used in the model training paradigm for which different training techniques were explored including domain-adaptive pre-training and LoRA fine-tuning. As there were no corresponding scratch notes in the dataset, few-shot prompting with a human-in-the-loop evaluation process was used to generate matching scratch notes. The historical progress notes and generated point-form notes were used to fine-tune Llama 2 and 3 models on the desired task. Different models' outputs were evaluated and compared before the final model, a fully fine-tuned Llama 3 8B Instruct model, was selected for a pilot study at KidsAbility in which the custom model was compared against the proprietary Microsoft Co-Pilot model. Ten OT's participated in the study, using Co-Pilot and then the custom model to write their progress notes for three weeks each.

It was found that providing training on how to most effectively use the custom model is important in reducing the amount of time spent on the process. After training, the average time taken to write a note was 7.6 minutes compared to an average of 13.8 minutes before training, both of which are based on subjective reporting. The progress notes written during the pilot study were also used in a quality assessment, in which four OTs scored the custom model notes, Co-Pilot notes, and manually written notes on multiple criteria. Results for this evaluation demonstrated that the notes written with the custom model were of high quality, receiving the highest score for three criteria and the second highest score for the remaining two. For all criteria, the custom model notes scored higher than the manually written notes. Objective timing data collection for determining the impact of using the custom model compared to not using any model was limited by the availability of clinicians.

Acknowledgments

I would like to thank Smile Digital Health for sponsoring this research and making it possible.

I also want to express my appreciation for those at Levio Business and Technology, who allowed us to use their AWS account for the first stages of data processing and model training.

Thank you as well to the research and innovation team at KidsAbility who initiated the project and provided invaluable insight and support along the way. Furthermore, thank you to all of the clinicians at KidsAbility who allowed us to shadow their work, ask many questions, and collect data while they worked.

At the University, I would like to thank my lab mates and fellow graduate students for bringing so much joy to my time here, as well as my supervisor, Bryan Tripp, for his guidance and support throughout my degree.

Finally, I would like to thank my friends, my family, and my partner, Kiernan, for their love and support.

Table of Contents

Author’s Declaration	ii
Statement of Contributions	iii
Abstract	iv
Acknowledgments	v
List of Figures	ix
List of Tables	x
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	2
1.3 Contributions	3
2 Background	4
2.1 Occupational Therapy Documentation	4
2.2 Large Language Models in Healthcare	5
2.2.1 Natural Language Processing	5
2.2.2 Transformers	5
2.2.3 State-of-the-Art	8
2.2.4 Applications in Healthcare	10
2.2.5 Domain Adaptive Pre-training	11
2.2.6 Fine-Tuning	12
2.2.7 Model Inference	13
2.2.8 Synthetic Data Generation	14
2.3 Amazon Web Services	15
2.3.1 Amazon S3	15

2.3.2	Amazon Comprehend	16
2.3.3	Amazon SageMaker	16
3	Datasets	17
3.1	Overview	17
3.2	Preparing Datasets	18
3.2.1	Goldcare Datasets	18
3.2.2	AlayaCare Dataset	19
3.3	Redaction Process	20
4	Generated Scratch Notes	23
4.1	Overview	23
4.2	Methods	24
4.2.1	Generation Procedure	24
4.2.2	Evaluation of Generated Data	25
4.3	Results	28
4.3.1	Evaluation Stage 1	28
4.3.2	Evaluation Stage 2	33
4.3.3	Evaluation Stage 3	35
4.3.4	Generation Procedure Used	36
4.4	Discussion	36
5	Model Training	39
5.1	Methods	39
5.1.1	Domain Adaptive Pre-training	40
5.1.2	Fine-tuning	40
5.1.3	Model Evaluation	41
5.2	Results	43
5.2.1	Pre-trained Models	43
5.2.2	Fine-Tuned Models	46
5.3	Discussion	55
5.3.1	DAPT and Catastrophic Forgetting	55
5.3.2	LoRA vs. Full Fine-tuning	56
5.3.3	Llama 2 vs. Llama 3	56
6	KidsAbility Pilot Testing	58
6.1	Methods	58
6.1.1	Pilot Overview	58

6.1.2	Clinician Note Scoring	59
6.1.3	Timing Data Collection	60
6.2	Results	60
6.2.1	Qualitative Results	60
6.2.2	Comparing Unedited and Edited Model Notes	66
6.2.3	Clinician Evaluation of Generated SOAP Notes	66
6.2.4	Timing Results	68
6.3	Discussion	69
6.3.1	Importance of Training Clinicians to use AI tool	69
6.3.2	Editing the SOAP Notes	70
6.3.3	Generated SOAP Note Quality	70
6.3.4	Comparing Timing Data	71
7	Conclusions	74
7.1	Summary	74
7.2	Limitations	75
7.3	Future Work	76
	References	78
	APPENDICES	87
A	Generated SOAP Notes	88
A.1	Full Set of Generated Progress Notes	89

List of Figures

3.1	Process of preparing the GoldCare Dataset.	19
3.2	Process of preparing the AlayaCare Dataset.	20
4.1	Box and whisker plots showing the effect, from top to bottom, of different SOAP notes, instruction sets, and few-shot example pairs on the scores assigned to the generated scratch notes.	29
4.2	Box and whisker plot showing the effect of the instruction set used on the scores assigned to generated scratch notes in Evaluation Stage 2.	34
4.3	Prompt structure for ranking generated scratch notes.	35
4.4	Prompt structure for scratch note generation using instruction set 6.	37
5.1	Training loss curve for pre-training of Llama 2 over 1 epoch.	43
6.1	Rubric used to evaluate and score SOAP notes.	59

List of Tables

4.1	Rubric used by KidsAbility clinicians to score generated scratch notes. . . .	26
4.2	Rubric used by research team to score generated scratch notes.	27
4.3	Effect of SOAP note ID on Evaluation Stage 1 scores.	30
4.4	Effect of instruction set ID on Evaluation Stage 2 scores given to scratch notes.	33
4.5	Evaluation Stage 3 scores given to scratch notes averaged across multiple outputs from groups with the same parameters used for generation.	36
4.6	Duration and cost of final generations for GoldCare and AlayaCare datasets.	36
5.1	DAPT details and hyperparameters.	40
5.2	Fine-tuning training details and hyperparameters.	41
5.3	Training loss, evaluation loss, and total runtime for different fine-tuned models.	47
5.4	The number of votes for each generated response from the Llama 2 vs. Llama 3 fine-tuned models based on perceived quality.	48
6.1	Mean and standard deviation of ROUGE scores comparing custom model notes that are unedited and edited by clinicians.	66
6.2	Mean scores assigned to SOAP notes within each note type category. The highest mean score for each criterion is bolded.	67
6.3	Standard deviation of the scores assigned to SOAP notes averaged across the different scorers within each note type category. The lowest standard deviation for each criterion is bolded.	67
6.4	Quantitative timing data for writing progress notes with the custom model. Note ID indicates chronological order. Italicized entry indicates an outlier.	68
6.5	Quantitative timing data for writing progress notes manually. Note ID indicates chronological order. Italicized entry indicates an outlier.	69

Chapter 1

Introduction

1.1 Motivation

KidsAbility is a pediatric rehabilitation center with locations in Kitchener, Waterloo, and Cambridge. They provide services for children and youth that have specific communication, physical, or developmental needs. As such, they employ many speech language therapists, physiotherapists, and occupational therapists who provide services and treatment for KidsAbility’s clients. As in all areas of healthcare, an important aspect of a clinician’s job at KidsAbility when providing treatment and therapy is to write and maintain appropriate documentation for all clients. Documentation is essential for providing continuity of care, especially if a client needs to be transferred to a different clinician, for accountability and insurance purposes, as well as for strategic planning and progress reporting [3].

The research and innovation team at KidsAbility approached the Medical AI Research Lab with the idea of using artificial intelligence (AI) to make the process of completing documentation more efficient for their therapists. They were motivated to do so by the fact that, while documentation is very important, it is also a very time-consuming process for clinicians that can limit the number of clients that they are able to treat. By reducing the amount of time that a clinician spends writing documentation, KidsAbility projects that capacity for treatment can be increased. In Ontario, as elsewhere, there are long waitlists for occupational therapy treatment for children and youth, so increasing capacity is an essential goal for pediatric rehabilitation centers like KidsAbility so that they can better meet the needs of their communities. In particular, we chose to focus on reducing documentation time for their occupational therapy (OT) program.

The process of completing documentation and other administrative tasks can also be repetitive and tedious for the clinicians, including the occupational therapists, reducing job satisfaction [14]. An example of a documentation task that clinicians perform very frequently is turning the point-form notes taken during or immediately after treatment visits into a properly formatted and more complete SOAP note. These progress notes, written in SOAP format, are central to OT treatment, and help to track and assess progress. However, internal investigations at KidsAbility have found that this process can take up to 20 minutes or more for each appointment and therefore can account for a substantial fraction of a clinician’s working hours, or else may need to be completed outside of their working hours.

Current advances in large language models (LLMs) have been identified as having the potential to automate documentation tasks for healthcare professionals such as the clinicians at KidsAbility. The innovation team at KidsAbility explored using ChatGPT, an LLM that is available online, as a tool to help write SOAP notes based on basic information from an appointment, and were impressed by the results. However, a proprietary model such as ChatGPT may not be most appropriate for various reasons including privacy and security risks if the data contains personal health information (PHI). Additionally, ChatGPT may be overly large and require more computation than would otherwise be necessary for the task. A smaller, more specialized model could be run in a secure environment controlled by KidsAbility, and would have less of an environmental impact than the much larger proprietary models. Furthermore, it could be trained to have specific pediatric OT domain knowledge.

1.2 Objectives

The goal of this thesis is to develop a custom language model that can be used by occupational therapists at KidsAbility to complete their progress notes in less time than their manual process currently takes without reducing the quality of the notes. The model is intended to transform point-form scratch notes from OT treatment appointments into a draft of a progress note in SOAP format that the clinicians can edit and use. Furthermore, the custom model is intended to be smaller in size than the large proprietary models available and private such that it can be hosted on servers in KidsAbility authorized environments and data that potentially contains PHI does not have to be shared with third party companies.

Given these goals, we chose to fine-tune the open-source Meta Llama language models that are comprised of 7 to 8 billion parameters on this specific task. Fine-tuning on

historical data from KidsAbility was performed in order to train the model to produce text that sounds like OT clinicians and that uses domain terminology appropriately.

1.3 Contributions

The primary result of this work is a custom language model that has been piloted by OT clinicians at KidsAbility to assist with the documentation process by transforming scratch notes into draft SOAP notes. Additionally, results from the pilot study at KidsAbility can provide insight into the use and application of such language models in the area of pediatric rehabilitation. A complete summary of the contributions of this thesis is as follows:

- Two processed datasets of historical progress notes in SOAP format from which incidental PHI has been redacted using Amazon Comprehend and custom code. The datasets were shared with KidsAbility rather than with the broader research community. Chapter 3 details the composition, processing and redaction of the datasets.
- Two datasets of generated point-form scratch notes that correspond to the historical progress note datasets. Chapter 4 describes the process of generating these datasets with the open-source Llama 2 70B Chat model including the human-in-the-loop evaluation process that was undertaken to ensure the generated data was of sufficient quality.
- A family of Llama 2 models that underwent different training paradigms including domain-adaptive pre-training, full fine-tuning, and LoRA fine-tuning as well as a final fine-tuned Llama 3 model. The training methods and results of the model training process are covered in Chapter 5,
- The findings from a pilot study at KidsAbility that provide insights into the use of the custom model in a clinical setting. Qualitative and quantitative results from the study are discussed in Chapter 6.

Chapter 2

Background

2.1 Occupational Therapy Documentation

Pediatric occupational therapy (OT) focuses on enhancing the daily activities and participation of children and youth through evidence-based interventions. Key areas include self-care, play, academic performance, social participation, and overall mental and physical health [4, 8]. OT approaches aim to improve the functional skills, health routines, and overall well-being of children and youth, allowing them to fully engage in their environments.

Progress notes are written by OT's to record treatment progress and outcomes, and are required for all treatment appointments. Such documentation is essential for maintaining complete and accurate records that can be used to monitor patient progress, support clinical decision making, and demonstrate the effectiveness of treatment [3]. Additionally, documentation is essential for legal protections given that incomplete or inaccurate records can negatively affect quality of care and have serious legal implications.

At KidsAbility, clinicians are asked to write their progress notes in a standardized format that is common in the medical community, called SOAP format, where SOAP stands for Subjective, Objective, Assessment, and Plan [29]. The Assessment section is also sometimes referred to as the Analysis section. Under the Subjective heading, clinicians write out what is reported by the client, or what the client's parent reports in many cases for pediatric occupational therapy. Clinical observations are documented under the Objective heading, and may include, for example, descriptions of the results of trials with different occupational therapy equipment or tools. The Assessment or Analysis section is

where a clinician writes the conclusions that they have come to based on the Subjective and Objective contents. Finally, under the Plan heading, a clinician writes about any additional tests or steps that are required or suggested, such as following-up or taking a break from practice. The SOAP note format is intended to help clinicians organize all essential information so that the progress note is succinct, complete, and easy to follow [29].

2.2 Large Language Models in Healthcare

2.2.1 Natural Language Processing

Natural language processing (NLP) is a field that focuses on the interaction of human users with computers through natural languages, such as written and spoken English. Within NLP, common tasks include speech recognition, machine translation, text classification, and text generation. The overarching domain of NLP can be broken down into the subcategories of Natural Language Understanding (NLU), which includes a computer’s ability to analyze and derive meaning from text or speech, and Natural Language Generation (NLG), which focuses on producing text or speech based on previously given data [12]. An example of an application of NLU is sentiment analysis wherein the computer is tasked with classifying language data by sentiment (e.g., positive or negative). NLG is utilized for applications such as chatbots and conversational agents that respond to given language input and engage in dialogue with users.

Modern NLP for both NLU and NLG applications typically involves machine learning, in which models are trained on large corpora of text to perform the NLP tasks. In order for an NLP model to work with natural language, tokenization is essential [24]. Tokenization involves systematically breaking up text into smaller, structured units called tokens that can be processed by the machine learning models. Typically, tokens roughly correspond to words, but can also be part of a word [24].

2.2.2 Transformers

Transformers, introduced by [39], are hugely influential in the area of NLP and currently exhibit the best performance for various language tasks. The transformer architecture improves on previous models used for language tasks, such as recurrent neural networks (RNN) and convolutional neural networks (CNN), by using self-attention mechanisms to

process large input sequences in parallel while encoding long-range dependencies and context [39]. RNN models were designed to handle sequential data, such as words or tokens in a sentence. Hidden states in an RNN are influenced by previous steps in a sequence, capturing temporal dependencies [37]. The main drawbacks of RNNs are that their model architecture does not allow for parallelization, which limits their computational performance, and that long-range dependencies or context can be lost due to vanishing gradients [39]. CNN models, on the other hand, are highly parallelizable neural network models that apply convolutional filters to input data in order to capture local patterns. CNNs are widely used for applications where the input data is in the form of images, but when used with text, CNN models struggle to capture longer-range dependencies without needing very deep architectures [39].

A key aspect of the transformers architecture is the application of self-attention mechanisms, which allow the model to create context-aware representations of each word by looking at the relationships with the other words in the sequence [39]. To compute attention scores, the tokens in the sequence are each converted into embedding vectors (x_i) within some continuous vector space, and then multiplied with three learned weight matrices (W_Q , W_K , W_V) to obtain query, key, and value vectors.

$$\begin{aligned} q_i &= x_i W_Q, \\ k_i &= x_i W_K, \\ v_i &= x_i W_V \end{aligned} \tag{2.1}$$

With the query, key, and value vectors for two different word embeddings, an attention score is calculated using the dot product of the query vector of the first embedding (Q_i) and the key vector of the second (K_j), divided by the square root of the key vector's dimension (d_K). Then, a softmax function is used so that all attention weights for the first embedding sum to 1. Finally, the output for each word embedding is determined by taking the weighted sum of the value vectors for all other tokens in the sequence, where the weight applied to each value vector corresponds to the attention weight calculated for that particular pair. Equation 2.2 shows this calculation, where Q , K , V represent matrices where the corresponding vectors have been packed together.

$$Attention(Q, K, V) = softmax(\frac{Q_i \cdot K_j}{\sqrt{d_K}})V \tag{2.2}$$

Computing self-attention is particularly computationally efficient and scalable as it makes use of fast matrix multiplication operations. Furthermore, transformers models

compute multiple sets of attention scores in parallel, using multi-headed attention. Multiple attention heads allow different types of features and relationships in the data to be captured, so that the resulting representations are richer and more nuanced. Each attention head learns different weight matrices, such that the key, query and value vectors for the same word embedding will be different across different heads. Each attention head computes its outputs independently, and then the outputs from all heads are concatenated along the feature dimension. Finally, this concatenated output undergoes a linear transformation using a weight matrix W_O .

Since transformers use only self-attention, which does not account for position in a sequence, it is necessary to specifically encode positional information into the input embeddings so that information on word / token order is accounted for. To this end, positional encodings are used, added to the input embeddings by summing the two together. Different types of positional encodings can be used including both learned and fixed embeddings. [39] use fixed sinusoidal positional encodings.

The overarching architecture of the transformer model introduced by [39] includes an encoder and a decoder stack. In each of these, the same general components are present, including multi-head self-attention sub-layers and position-wise fully connected feed-forward neural networks, as well as residual connections and layer normalization for each sub-layer. The decoder has an additional multi-head self-attention sub-layer that takes the output of the encoder as its input. In general, encoders process input embeddings to produce useful representations, while decoders generate output sequences using such representations. So, in the case of [39], when they are performing a translation task, the encoder takes the English text as input and creates the continuous representations of the words, and the decoder outputs the text in the other language such as German or French.

Different transformer models have encoder-decoder architectures, but also encoder-only or decoder-only architectures. An example of a transformer model that uses an encoder-only architecture is the Bidirectional Encoder Representations from Transformers (BERT) model, introduced by Google in 2018 [10]. The BERT architecture can be used for a wide range of NLU tasks, including text classification, and question-answering where the model is able to provide a text sequence from a document as an answer to a natural language question. One way in which the BERT model is hugely influential, is that it introduced the idea of fine-tuning an already trained model to optimize model performance on a specific task, often in a specific domain. The concept of fine-tuning will be explored further in Section 2.2.6.

Meanwhile, generative language models that perform NLG tasks such as conversational agents, typically have decoder-only architectures that focus on generative capabilities to

produce new language content. An example of decoder-only models are OpenAI’s Generative Pre-trained Transformers (GPT) models. The original GPT model was introduced in 2018, and consisted of multiple transformer decoder blocks with self-attention and feed-forward neural network sub-layers [31]. The GPT model, like other many other transformers models, was trained in an autoregressive way, such that it was given all previously generated tokens, and tasked with generating the next token. This type of training is often referred to as Causal Language Modeling (CLM) and is used to pre-train generative models on text corpora [17]. CLM is described in further detail in Section 2.2.5.

2.2.3 State-of-the-Art

More recent advances have largely been driven by the scaling of transformer models to larger sizes, with the term large language model being introduced to refer to models with hundreds of millions or billions of parameters. In this context, parameters refer to the various numerical values that are learned in the model from training, such as the weight matrices of an attention head, with more parameters allowing for a higher capacity for understanding and generating natural language. [19] established a scaling law for LLMs and demonstrated that increasingly large model sizes and datasets exhibit increasingly high performance and generalizability. In addition to scaling the models, other advancements have been introduced such as reinforcement learning from human feedback (RLHF), which improves LLM outputs by incorporating human evaluations [7], and the creation of multi-modal models that integrate images [30]. A particularly relevant early advancement is also the ability to fine-tune already trained models on specific tasks that may have been outside of the scope of their original training [10].

Current state-of-the-art models include OpenAI’s GPT-4 [26], and the most recent Llama models from Meta [38, 11]. GPT-4 is a more recent iteration of the original GPT model, released in 2023, that maintains aspects of the original decoder-only architecture, while being larger, with hundreds of billions to trillions of parameters trained on massive, diverse text corpora and utilizing more advanced techniques for training. The exact number of parameters and training paradigm for GPT-4 are proprietary. OpenAI also has released versions of the GPT models fine-tuned for chat applications, using RLHF among other training methods, collectively called ChatGPT. ChatGPT, which used earlier GPT models when it was first made available to the public in 2022 through a web application, has been hugely influential in society, impacting communication, education, and creativity, among other aspects of modern life [16]. Arguably the biggest breakthrough in AI that ChatGPT represents is based in how user-friendly it is, having been trained to produce human-like, readable text and to respond effectively to natural language prompts and instructions.

However, from a research perspective, the application of the GPT models is somewhat limited by the proprietary nature of the models, such that direct access and customization of the model weights is restricted, and fine-tuning is only possible through OpenAI’s API.

The most recent of Meta’s Llama models, Llama 3, was released in April 2024, with additional models (Llama 3.1) released 3 months later in July, while Llama 2 was released the year before in 2023 [11, 38]. All of Meta’s models are open-source and, therefore, made available to the public through platforms like HuggingFace, Amazon Web Services (AWS), and Google Cloud. Both Llama 2 and 3 model releases included different sizes of the respective models. Llama 2 encompasses three models in three sizes including 7B, 13B, and 70B parameter models, while the Llama 3 model release in April includes 8B and 70B parameter models. The smaller models offer more versatility to potential users in terms of training and deployment as they require less computational power than the larger models to use. This ties into the open-source design philosophy of the Llama models, as it allows for greater customization, and potential for innovation from researchers and others in the development community who can use cloud computing, for instance, to work with these models in a more economically plausible way. Similarly to ChatGPT, the Llama models of all sizes have been fine-tuned to interact with users in a conversational manner through the process of fine-tuning and other post-training techniques such as RLHF and direct preference optimization [38, 11].

Comparing the GPT models and Llama models, the GPT models are larger and more capable, but Llama 3 in particular offers very impressive capabilities with far fewer parameters, especially in the case of the 8B parameter models. The smaller size of the Llama models makes them more accessible for research and industry applications that involve further fine-tuning. The training procedure for the Llama 2 models involved extensive pre-training via CLM on 2 trillion tokens, supervised fine-tuning on a comparatively small, carefully curated dataset to train the model to respond to chat style dialogue and instructions, and then RLHF was employed to align the model with human preferences and instruction following [38]. While the same general training steps were taken to train Llama 3, Meta leveraged improvements in pre-training and post-training to improve the performance of Llama 3 compared to Llama 2 in terms of benchmarks such as MMLU, GPQA, and human evaluation [11]. For example, Llama 3 was trained on significantly more tokens that went through a more rigorous quality assurance process and was trained with a training paradigm that prioritized complexity management. The post-training or fine-tuning procedure was kept simple, including only basic supervised fine-tuning, rejection sampling, and direct preference optimization as opposed to more complex RLHF algorithms [11].

2.2.4 Applications in Healthcare

There are several applications for LLMs in healthcare that have been explored in the literature, including clinical decision support, patient engagement, medical research, and medical education [43]. The ability of LLMs to analyze patient data and generate differential diagnoses, for example, can be used by doctors to enhance decision-making, while the ability of chatbots powered by LLMs to translate medical findings into layman’s terms can be very valuable for patient support and education. In medical research, LLMs can digest large corpora of medical literature and provide readable summaries for researchers to review. An area in healthcare that is particularly transformative for clinicians, though, is the application of LLMs in the domain of clinical documentation, such as automating medical coding, automating transcriptions, and generating clinical reports and notes [43].

Clinical report generation requires LLMs to generate new text, summarize any relevant findings, and / or retrieve information from existing sources. Some clinical report generation is also multi-modal, using vision models that provide the LLMs with annotations from medical imaging so that they are able to generate reports using both textual and visual inputs [40]. Since the release of ChatGPT there has been significant interest in using LLMs to create and manage clinical progress notes and discharge summaries [27, 25]. Specifically, the use of ChatGPT for the purpose of writing progress notes has been investigated in the literature and was explored by KidsAbility clinicians. [25] used ChatGPT to write progress notes for chronic pain assessment using varying prompts. Their preliminary tests led the authors to conclude that ChatGPT shows great potential for helping to generate healthcare progress notes, but that this relies heavily on the effectiveness of the prompting techniques and the accuracy of the data that is provided as input. Furthermore, the authors stress the potential for inherent biases, data privacy concerns given that health data is being submitted to a proprietary model, and the need for healthcare practitioners to maintain accountability for their own notes. KidsAbility clinicians similarly expressed the importance of prompting when using ChatGPT to help them write their progress notes, reporting that much of the time they spent on the task of producing occupational therapy SOAP notes was in refining the instructions that they gave to the model. The clinicians corroborated the need for carefully considered input data as well, often writing out very high levels of detail in their scratch notes that they gave the model as input data. In fact, clinicians reported minimal time savings in using ChatGPT to generate SOAP notes as a result of the need for such prompt engineering and such high levels of details in the input notes.

While using LLMs for clinical report and notes generation has many potential benefits, primarily with regard to time savings and also potentially in terms of increasing the quality

and consistency of progress notes, there are also challenges and ethical considerations to be aware of [43]. Privacy and security of personal health information (PHI) is an essential ethical consideration when using such tools. The proprietary nature of models such as ChatGPT makes this an especially relevant concern, as data provided to the models can be saved and used by the third party company. Using open-source models such as the Llama models deployed on private servers can mitigate this risk. Additionally, there are concerns of accuracy and the potential for LLMs to hallucinate, which is when the model generates plausible but incorrect text. Hallucinations or gaps in the clinical reports written by LLMs could have serious negative consequences for patient care. Therefore, it is essential that clinicians carefully review the generated notes. Such review can be a time-consuming process.

A lack of appropriate evaluation methods for LLMs in this application also poses practical and ethical challenges. Compared to other tasks, clinical report generation is not well-suited to automatic evaluation metrics that are typically used such as BLEU, ROUGE, and METEOR, as the task is generative in nature, and the content of clinical notes is highly specialized [43]. Current methods focus on lexical metrics, and provide limited insight into the accuracy of the clinical text and the potential risks associated with the LLM [41]. There is a need for evaluation methods that assess more than surface-level text similarity and that focus on the medical context. It is likely that such evaluation methods will require human insight and evaluation, with the requirement that the human evaluator is knowledgeable enough in the domain to score the LLM’s responses based on accuracy, relevance, fluency, and more [5].

2.2.5 Domain Adaptive Pre-training

Large language models are pre-trained on massive text corpora. In the case of Llama 2, this includes 2 trillion tokens of data from publicly available sources. The process of pre-training used for generative LLMs such as the GPT or Llama models is a self-supervised process called Causal Language Modeling. In CLM, the transformer model is trained on text such that it is tasked with predicting the next token in a sequence in an auto-regressive manner [17]. When the model predicts a token that is not correct based on the reference text it is using, the calculated cross entropy loss is used to adjust the model parameters so that, over time and many examples, the model learns to represent the conditional probability of word sequences and thus to be able to generate coherent and context-specific text.

Domain adaptive pre-training (DAPT) is a form of transfer learning that involves continued pre-training on text from a specific domain, such as scientific literature or, as is the

case for this project, clinical progress notes from OT appointments. DAPT is used to train a general pre-trained model to understand domain-specific vocabulary and nuances and to produce text more suitable to the target domain [15]. As it is continued pre-training, the same process of CLM can be used to train an LLM on the new text corpora, with all of the model parameters being updated as it trains on the new corpora.

2.2.6 Fine-Tuning

Similarly to DAPT, fine-tuning is a process that adapts a pre-trained LLM to be useful in a more specific context. In the case of fine-tuning, a smaller, more specific dataset is used to adapt a model to a particular task through further, specialized training. Fine-tuning allows a model to leverage the general knowledge and language modeling capacity that it learned in pre-training, while being specialized for a task such as generating dialogue or generating clinical reports. The paradigm of fine-tuning a general pre-trained language model was introduced in [10] and has been hugely influential in allowing other researchers and developers to use and adapt general models to perform a large variety of tasks in many domains. One of the reasons that fine-tuning is so important is that very few entities have access to the resources that are required to pre-train a language model, let alone an LLM, from scratch. Fine-tuning can allow developers to customize a model for a particular purpose on a practical budget and within a reasonable time frame as the model has already undergone general pre-training and has general language modeling capabilities.

Cloud computing, such as the resources available through Amazon Web Services, helps to make fine-tuning models in this paradigm more accessible by providing access to the necessary computing resources on an as-needed basis, and therefore in a more affordable way. Furthermore, recent advances in parameter efficient fine-tuning (PEFT), allow researchers to use less powerful computing resources to fine-tune LLMs. QLoRA is a prevalent PEFT method that leverages quantization and Low Rank Adapters to train a large model very efficiently in terms of both computation and the size of the fine-tuning dataset [9, 18]. Quantization allows the pre-trained model to be represented and fine-tuned in a new data type, called 4-bit NormalFloat, as opposed to the standard 16-bit training. This significantly reduces the size of the model for training, so less memory is required on the GPUs being used for training. Furthermore, the Low Rank Adapter is a small set of learnable weights that is added to the pre-trained model and that is tuned, while the pre-trained model parameters remain constant.

The specific process or algorithm used for fine-tuning varies depending on the task that the model is being adapted for. In the case of this project, a specific dataset that

represented the task of creating SOAP notes from scratch notes was required. With such a dataset, the fine-tuning task could be framed as a CLM task such that the model learned to use the scratch note as context from a prompt, and then generate the corresponding progress note in the autoregressive manner characteristic of CLM, predicting one token at a time. This can be considered a form of supervised fine-tuning as the dataset has explicit examples of how the model should perform its task. QLoRA fine-tuning, and full fine-tuning where all model parameters were updated during training were both used to train models for comparison. However, the quantization aspect of QLoRA was not used, and therefore the model training paradigm used is referred to as LoRA only.

2.2.7 Model Inference

Model inference is the process of using models to make predictions or generate outputs based on provided input data. A trained LLM can be used to make inferences through various text generation interfaces (TGI), such as HuggingFace’s TGI available in Python code through the Transformers library. Using a TGI, users can provide an LLM with a text prompt that is tokenized and provided to the model. The model is then tasked with next token prediction to generate its response. The model’s response is returned to the user by the TGI in plain text. An example of a user-friendly TGI that has additional functions and capabilities is the web app through which ChatGPT is accessed.

In model inference, there are key parameters that can control aspects of the model’s response. One such parameter is called temperature, which controls the randomness of the output by adjusting the probability distribution of the predicted tokens [28]. Lower temperatures adjust the probability curve to be narrower and thus favor the highest probability tokens only, so that the model produces more deterministic outputs. Conversely, higher temperatures flatten the probability curve, increasing the likelihood that the model will predict tokens that are less probable, which means that the generated text will be more creative and variable. For instance, with higher temperatures, a model is more likely to generate very different outputs with the same prompt [28]. Users of TGI’s that are performing model inference can adjust the temperature of the generation in order to control the balance between coherence and variability in the model’s output.

Additionally, prompting plays a crucial role in model inference for LLMs. A prompt is an initial text input or instruction that serves as the context for a model’s generation process. Prompts can be as straightforward as a question or a simple, one sentence instruction, or as elaborate as a complex set of instructions for what to generate and how to present the generated text. System prompts are used to define the role of the LLM and its behavior

including rules and constraints for the model’s generations. For example, a system prompt may tell the model that it should act as an assistant, should use an academic tone, or must not include any content that is harmful or toxic in nature.

Generally, prompts are most effective when they are clear, concise, and have minimal ambiguity for the model to interpret. Meta’s prompting guide provides tips for writing prompts for the Llama models along these lines [23]. The guide recommends providing explicit instructions regarding the desired outputs, using formatting like enumerated lists of rules, and providing examples of inputs and outputs. Often, it is necessary to experiment with different prompts in order to find one that is most effective at generating the desired outputs.

When examples of inputs and outputs are provided in the prompt, this is referred to as few-shot inference, referencing the fact that the model is making an inference based on the context of the examples. Zero-shot inference, on the other hand, is when the model is given only a set of instructions to complete a task without any examples [36]. Typically, the extra context provided with few-shot inference helps the model to perform better on the desired task than it would with zero-shot inference.

2.2.8 Synthetic Data Generation

When there is insufficient, incomplete, or imbalanced data for training on a specific task, it is possible to use LLMs to generate textual data to augment the existing dataset or create a synthetic dataset. The augmented or synthetic dataset can then be used to fine-tune a model on the task. Researchers have explored this application of LLMs in recent years, finding mixed results in the fine-tuned model’s abilities that are dependent on the quality of the generated data [20, 21, 22]. Important aspects to consider to generate high quality data include realism, accuracy, relevance, and any potential biases that could appear in the data. As the evaluation of these aspects is very difficult to automate, typically human review is required to evaluate the generated examples. Human-in-the-loop approaches involve such human review to curate the dataset, improving its quality and the benefits of fine-tuning on the dataset.

An example of a technique that can help to enhance the quality of generated data is using few-shot inference as opposed to zero-shot inference, requiring a set of high quality examples of the task for the model to use as context. Typically this small set of examples would be provided by human domain experts. In particular, larger models, such as GPT-3.5 or GPT-4 and Llama 70B, that have been fine-tuned to follow instructions demonstrate good performance in few-shot inference and are able to mimic the style and presentation of

provided examples to generate new data [2]. In the case of this project, few-shot learning was used to generate scratch notes, a succinct and short-form version of the progress note, given a real example of a historical SOAP note.

2.3 Amazon Web Services

AWS is a cloud computing platform that offers a comprehensive set of software services for businesses, researchers, and individuals and that has had a profound impact in the field of machine learning and LLMs. It offers access to powerful computing resources that can be used to train and deploy models in a scalable, secure, and cost-effective manner. Additionally, its services are widely used for data storage and management.

In this project, a third-party contractor, Levio, used AWS to securely store and transfer medical records from one Electronic Medical Record (EMR) system to another. The contract to perform the transfer of data was created without this project in mind and was instead intended to allow KidsAbility to use a new EMR system that they felt would better suit their needs. However, the opportunity to use the data while it was in the process of being transferred was identified. Thus, the data was transferred from the GoldCare EMR system to the AlayaCare EMR system by the consultants at Levio and they were also able to provide the research team with access to relevant data in the process. The research team was then able to use AWS’s large set of services to prepare and process the data, as well as train and deploy the models within a secure cloud environment owned and authorized by KidsAbility.

2.3.1 Amazon S3

Amazon S3 (Simple Storage Service) is AWS’s scalable object storage service, which allows users to store, retrieve, and manage large amounts of data. S3 is designed for durability, with data replicated across multiple Availability Zones, ensuring reliability. It’s also secure, offering features like encryption and fine-grained access control through Identity and Access Management policies [33]. Therefore, S3 was used throughout the project to store and access all data and model artifacts.

2.3.2 Amazon Comprehend

Amazon Comprehend is an NLP service available on AWS with pre-trained models for performing specific tasks [32]. For example, it can be used for the automatic detection and redaction of PHI from text. With automatic redaction of PHI, it is feasible to anonymize a large corpus of text in a time efficient manner and in such a way that the user never comes in contact with the PHI itself. This made the service valuable for the research team for the redaction of any incidental PHI from the given data.

2.3.3 Amazon SageMaker

Amazon SageMaker is a machine learning service that enables data scientists and developers to build, train, and deploy machine learning models at scale [34]. SageMaker offers features such as notebooks, support for custom code in popular frameworks including PyTorch and HuggingFace as well as the hosting of inference endpoints for real-time predictions and training job endpoints that use distributed infrastructure.

In particular, SageMaker’s notebooks were used extensively in the project to develop code for data analysis and processing, provide code-mediated access to endpoints, and analyze results. All of these tasks were performed in a `ml.t3.medium` notebook. Additionally, two types of endpoints were used, including a training job endpoint and inference endpoints that were used for data generation using few-shot prompting and then for evaluation and deployment of the custom language models.

Chapter 3

Datasets

3.1 Overview

To train a model to write progress notes in SOAP format that reflect the style of the clinicians at KidsAbility, it was necessary to use representative examples for training. The ideal dataset would have consisted of historical SOAP notes linked to corresponding scratch notes. However, it is not standard practice for clinicians to save any point-form notes they take before writing out the SOAP note, so there were not any historical scratch notes or equivalent data to use for this purpose.

KidsAbility did have access to tens of thousands of historical SOAP notes stored in their EMR system. Prior to 2023, the EMR system used at KidsAbility was GoldCare and all historical data was located in the GoldCare system. In 2023, KidsAbility transitioned to a new EMR, AlayaCare. They contracted Levio, a digital consulting firm, to facilitate the migration of all historical data from GoldCare to their new EMR system using AWS. As part of this process, Levio and KidsAbility agreed to provide the research team with access to the historical data via AWS. It was agreed that the research team would complete all development work on the model within the secure AWS environment as well. Levio provided data from a ten year period that they ensured was de-identified such that no PHI was linked to the note content.

As the project progressed, it was decided that newer data from AlayaCare that had been collected after the project's beginning stages, would be more suitable for fine-tuning the large language model. This decision was based on the realization that the older historical data was not representative of the desired format for progress notes, as it was not sectioned

appropriately. Specifically, the Subjective, Objective, Analysis, and Plan sections of a traditional SOAP note were not applied consistently and the notes often consisted of two to four paragraphs of text without these headings. Therefore, KidsAbility provided the research team with access to more recent data from AlayaCare, which specifically enforces the four sections with its template. The newer data was collected between January 2023, when AlayaCare first began to be used, and March 2024, which was just prior to the training of the final models.

3.2 Preparing Datasets

3.2.1 Goldcare Datasets

In total, Levio provided 411,812 progress notes from the GoldCare EMR, including notes taken from 2012 to 2023. This time period of approximately ten years was chosen so that there would be a manageable number of progress notes for Levio to create a dataset from, but also because as the notes become older, they are also less likely to conform to the current desired SOAP format. The notes were de-identified by Levio such that any PHI linked to the notes, such as the client’s name and age, were excluded from the metadata. There was still a risk of incidental PHI being present in the note content though, such as the first name of the client.

All the progress note data was provided in a parquet file through AWS S3 resources. The dataset processing for training was conducted in notebooks on Amazon SageMaker within the Levio AWS account. Firstly, the data was filtered to include only notes from OT appointments, excluding the speech language therapy and physiotherapy notes. After this filtering, the dataset contained 37,437 OT notes.

To complete the remaining processing steps, the Llama 2 7B tokenizer was used to determine the token length of all the OT progress notes. The dataset was then filtered to include only notes with 50 or more tokens and fewer than or equal to 1000 tokens. To determine the token range to be used, the distribution of token length for all OT notes was plotted, and several examples from the upper and lower bounds were examined. The range of 50 to 1000 tokens was chosen based on the distribution, as it accounted for a majority of the notes. Additionally, the notes under 50 tokens were too brief to be considered true SOAP notes, while the notes that were longer than 1000 tokens could have caused difficulties in training depending on the model being used due to the limited context length.

The dataset was then divided into the pre-training dataset, meant to be used for domain adaptive pre-training, and the fine-tuning dataset. As fewer examples are necessary for

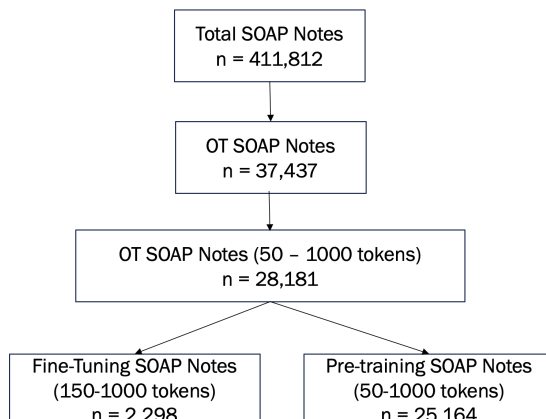


Figure 3.1: Process of preparing the GoldCare Dataset.

fine-tuning than for domain adaptive pre-training, approximately 10% of the data was used to create the fine-tuning dataset. The data was divided using the client ID, which was a random ID associated with each set of a particular clients' notes. Client ID's that made up 10% of the data without having overlapping client entries in the remaining 90% were used to divide out the fine-tuning dataset. Checks were performed to ensure that the data in the fine-tuning dataset was randomly distributed across the full time range.

Further filtering was performed on the fine-tuning dataset after further exploration of that dataset. The dataset was filtered to include only notes with at least 150 tokens. Notes that had fewer than 150 tokens were excluded because these were found to be far more likely to not conform to the desired SOAP format. Often, the notes between 50 and 150 tokens were emails or brief notes about missed appointments. After this additional filtering, the fine-tuning dataset consisted of 2298 notes. The pre-training dataset was not filtered further, and consisted of 25,164 notes.

3.2.2 AlayaCare Dataset

The AlayaCare dataset, which was collected during the earlier stages of the project, was de-identified and provided via AWS S3 by KidsAbility. This dataset consisted of 1,127 SOAP notes in total. Again, a combination of inspection of individual examples and overall token length distribution was used to determine an appropriate range to be filtered for. The Llama 3 8B tokenizer was used in this case to determine the number of tokens in each of the notes. A different tokenizer was used because in the time that had elapsed since

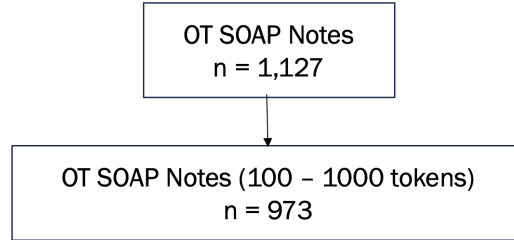


Figure 3.2: Process of preparing the AlayaCare Dataset.

processing the GoldCare dataset, Llama 3 had been released and was showing significantly improved metrics. Thus, the research team wanted to use this newly released model and its associated tokenizer for training with the newest data. We did not expect there to be a significant discrepancy in token lengths based on which of the two tokenizers was used, as they are very similar.

For the AlayaCare dataset, a token length range of 100-1000 tokens was selected. The minimum token length was smaller than in the previous dataset, both because there were fewer notes overall so we wanted to keep as many notes as possible, and because the new data was more reliably structured despite shorter token lengths. The more reliable structure is likely due to the fact that AlayaCare enforces a SOAP note template.

After filtering based on token length, the AlayaCare fine-tuning dataset comprised 973 notes in total.

3.3 Redaction Process

When working with healthcare data, it is important to be aware of any existing personal health information, and to take measures to safeguard the privacy and security of data that contain or may contain PHI. Any information within the data that could be feasibly used to link specific medical information such as treatment plans or a diagnosis with an individual is considered PHI. Thus, we had to take particular precaution with the SOAP notes provided in the dataset. Before the research team was provided with access to the

GoldCare or AlayaCare SOAP notes, all PHI that was externally linked to the note was removed. For instance, any metadata with the child’s name, age, or location was removed. Within the text of the SOAP notes, however, there was incidental PHI which could not be systematically removed before being provided to the research team. Incidental PHI in the SOAP notes primarily took the form of first names, including those of the client and those of the client’s parents or caregivers. Full names, including family or surnames, were not found in the data, though. There were also a few instances of ages being included or the names of schools. To estimate the occurrence of PHI in the notes, we took a random sample of 96 progress notes from the GoldCare dataset and skimmed the notes for names. 79 of the 96 names contained a first name of a client, parent, or other caregiver, so we estimated that first names occurred in the notes approximately 82% of the time.

Before training the model on the SOAP notes, or generating scratch notes, it was important to remove as much of the incidental PHI from the notes as possible. Primarily, this was because there is a risk with LLMs that they can be prompted to return portions of their training data [42]. KidsAbility also wanted the option to share the trained model with other related pediatric rehabilitation centers in Ontario, which they would only be able to if the model was not at risk of exposing any kind of PHI.

Amazon Web Services’ Comprehend was identified as a potential service that could be used to redact first names and ages in an automated manner. Comprehend is an NLP service that uses trained machine learning models to perform tasks such as identifying specific entities like names, and other private data. The redaction service provided by Amazon Comprehend identifies entities within the text and replaces these entities with placeholders that describe the entity type. For instance, if the model found a first name, “John”, Comprehend would return the text with “[NAME]” in place of this first name. The GoldCare and AlayaCare SOAP notes were run through the Amazon Comprehend service to replace any names and ages with [NAME], or [AGE] placeholders respectively.

After processing by Amazon Comprehend, the research team evaluated how well the service performed at redacting names by looking through a random sample of 100 SOAP notes. 26 out of 100 of the SOAP notes checked still had at least one instance of a first name appearing in the text, 17 of which were the client’s name. Therefore, it was estimated that the Amazon Comprehend Service had a success rate of approximately 75% for redacting names. Another key finding was that most of the names that were not redacted by Comprehend were names that were less common in Canada and North America, representing a bias in the AWS model missing the PHI of clients and families with less common names.

To address this gap in Comprehend’s redaction process, a list of first names for all clients

of KidsAbility was provided to the research team. To protect the privacy of KidsAbility clients, the list of first names was not linked to any other data. Using this list, the names could be programmatically removed and replaced with placeholders by using Python string utilities. The drawback of this method is that, unlike with Amazon Comprehend, the programmatic redaction process was not able to distinguish between the use of words such as “June” and first names with the same spelling. Thus, there were some examples identified where the name of a month or other proper noun would be replaced with “[NAME].” There were not any significant issues identified with this, however.

After both methods for redaction were used, we estimated the occurrence of first names in the data using the same random sample of 100 notes used for Comprehend testing. We found one instance of what appeared to be a client first name, and 3 instances of parent or other caregiver first names. Therefore our estimate for the success rate of client name redaction and parent or caregiver name redaction is 99% and 97% respectively.

Chapter 4

Generated Scratch Notes

4.1 Overview

The fine-tuning process for the model requires examples of the task being performed. In the case of this project, it was necessary to train the model on pairs of corresponding scratch notes and SOAP Notes. Ideally, the dataset for training would consist of the historical SOAP Notes and the scratch notes that therapists wrote out during their appointments. However, the process of taking scratch notes is not consistent across therapists, with some writing only a few words down on a piece of paper. Further, scratch notes are disposed of after the SOAP note is completed, so there are no historical scratch notes to use. Therefore, it was necessary to either ask KidsAbility clinicians to start saving their scratch notes and wait for the data collection to be completed over the course of months, or create synthetic data. The option of creating synthetic data to use in place of scratch notes was chosen because the data collection would be prohibitively time-consuming, and we wanted to be able to make use of the historical data that we had.

Data generation was a necessary part of the research project so that each historical SOAP note to be used for fine-tuning would have a corresponding synthetic scratch note. The Llama 2 70B Chat model was chosen for data generation as it could be deployed on AWS, and it is capable of generating text with specific parameters based on few-shot prompting. For few-shot prompting, the KidsAbility clinicians provided a dataset of 45 examples of scratch notes with their corresponding SOAP notes.

4.2 Methods

4.2.1 Generation Procedure

The Llama 2 model was deployed on AWS SageMaker using an inference endpoint attached to a ml.g5.48xlarge node, which has 8 NVIDIA A10G GPUs, with 24 GiB. The code deploying the SageMaker models and endpoints as well as the code for interacting with the model once deployed, was written in a SageMaker notebook. HuggingFace’s Text Generation Interface (TGI) was used for this purpose, because it allows the user to pass the prompt and text for inference and returns the model’s response in real-time. Both synchronous and asynchronous inference endpoints were used for scratch note generation. The asynchronous endpoint saved all of the generated data automatically to Amazon S3, while the synchronous endpoint simply returned the generated data to the notebook, such that the data had to be saved to S3 after generation was completed.

The larger Llama 2 model, with 70B parameters, was chosen for generation because its ability to perform few-shot learning was significantly better than the smaller models. To perform the few-shot learning, the 45 curated example pairs provided by the KidsAbility clinicians were randomly sampled so that the general prompt structure was as follows: one example of a SOAP note, the corresponding scratch note, a second example of a SOAP note, the second corresponding scratch note, and then a SOAP note with the instructions to create the new scratch note. Therefore the examples of SOAP note / scratch note pairs are used as examples for the model to know what a scratch note should look like in this context.

Various parameters for generation were tested to determine which produced the desired quality of scratch notes. The most significant of these parameters were the prompt instructions including the system prompt and the inline instructions. Other parameters that were tested were temperature, and whether or not a post-processing step was used. Two methods of post-processing were tested, including providing the model with the generated scratch note and asking it to make the note more concise, and providing the model with multiple generated scratch notes for the same SOAP note, asking it to rank the scratch notes based on a provided set of criteria.

All of the outputs from model generation were saved in Amazon S3, such that they could be retrieved again for analysis in Python using the SageMaker notebook.

4.2.2 Evaluation of Generated Data

To evaluate the scratch notes being generated, and to determine the best parameters to use for generation, it was necessary to use a consistent method that would capture the properties of scratch notes that are most important for training. Such properties primarily have to do with the realism of the generated data, since it is important for the purpose of fine-tuning that the synthetic data be as representative of the scratch notes that clinicians actually write as possible. To this end, a rubric for evaluation was developed in coordination with the KidsAbility research and innovation team. The rubric included three criteria for which it was necessary to have clinical expertise to evaluate the scratch notes, and one criterion that could be scored more objectively, and so was completed by the research team. The clinical criteria included Realism, Accuracy, and Distinction between Observations, Goals, and Subjective Reporting, and clinicians were asked to provide a score from 1 to 5 for each. The rubric can be seen in Table 4.1. In addition to these rubric criteria, the evaluating clinicians were also asked to mark any scratch notes that were particularly problematic in some way as “Red flag” scratch notes. The criterion that the research team was able to evaluate was called LLM Brevity and pertained specifically to the LLM’s performance regarding producing extraneous text. The rubric for LLM brevity is shown in Table 4.2. The goal for LLM Brevity, was for the model to produce the scratch note and no extra text, so the highest score was given to examples where there were no instances of the model including chat text such as “Notes” or greetings.

Three stages of evaluation were conducted with three different clinicians from KidsAbility. For each of these evaluation stages, the clinicians were provided with 80-100 examples of historical progress notes and the corresponding generated scratch notes. The clinicians filled out an Excel sheet based on the established rubric, scoring all of the generated scratch notes on the criteria. An additional column in the Excel sheets provided clinicians with the opportunity to “flag” any particularly problematic scratch notes, as well as a column for any written feedback they wanted to provide.

The first stage of evaluation focused on determining which prompt instructions provide the best scratch notes based on the established criteria as well as how influential the pairs of example scratch notes and SOAP notes are in determining the quality of the generated scratch notes. Five different, randomly selected historical SOAP notes were used, with eight different pairs of SOAP notes and scratch notes from the curated examples used for few shot learning, and five different prompt instructions. Two of the curated example pairs were used for each generation, so 4 sets of curated examples were used in total. Between the five SOAP notes, four example sets, and five prompt instructions, there were 100 examples to review in total. The 100 examples were organized based on which historical

Criteria	1	2	3	4	5
Realism	Does not resemble real Scratch Notes. It is obvious that it was generated.	Moderately dissimilar to real Scratch Notes. Could most likely recognize as generated.	Somewhat representative. Reasonable degree of similarity to real Scratch Notes.	Mostly representative. High degree of similarity to real Scratch Notes. Might be difficult to tell apart from real Scratch Notes	Representative of Scratch Notes that are consistently seen.
Accuracy	Includes many errors or significant error(s) in the text.	Includes potential errors or not fully accurate information.	Some small errors. Somewhat faithfully captures the content of the Progress Note.	No harmful errors in the text. Does a reasonable job at capturing the content of the Progress Note.	No errors in the text. Faithfully captures the content of the Progress Note.
Distinction between Observations, Goals, Subjective Reporting	Unclear when a point is a goal, observation or subjective report. Instances here would be misleading and problematic.	Moderately unclear distinction between goals, observations, and subjective reports. Could feasibly lead to confusion.	Somewhat clear distinctions. It is possible but somewhat unlikely that it could cause confusion.	Mostly clear distinctions. Very unlikely to cause a misunderstanding.	Clear distinction between goals, observations and subjective reporting.

Table 4.1: Rubric used by KidsAbility clinicians to score generated scratch notes.

Criteria	1	2	3	4	5
LLM Brevity	Output contains inexplicable extra text. Would need to be edited.	Output is not limited to desired Scratch Note. Ideally, some text would be removed before presenting to clinician.	The output contains some "chat" text. Somewhat less than ideal to present to clinicians.	The output is mostly free of unnecessary "chat" text. The output is mostly appropriate to present to clinicians.	The output contains minimal extra text. If this is what clinician's received as output, it would be perfectly acceptable.

Table 4.2: Rubric used by research team to score generated scratch notes.

progress note was used, so that clinicians could read the SOAP note once and then review the different generated scratch notes that correspond to that note. The generations were randomly ordered in relation to the prompt instructions and curated example sets and there was no indication of which was used in the spreadsheets.

The second stage of the evaluation focused on the prompt instructions, and whether or not post-processing, in the form of asking the model to make a generated scratch note more concise, was used. Four different sets of prompts were used, including two different prompt instructions for the initial generation, and a pre-processing prompt that was used with each. The same four curated example sets and five historical SOAP notes were used, resulting in 80 total generated examples. In the third stage of the evaluation, only the best prompt was used to generate scratch notes, but four scratch notes were generated for each of the five historical SOAP note and four curated example pairs, to test how much the quality of generated scratch notes varied with the same prompt and few-shot examples. Therefore, the third stage also included 80 generated scratch notes. Additionally, in this last stage of evaluation, the clinician was asked to rank the four scratch notes that were generated from the same prompt and curated example set from best to worse (1-4). The clinician's rankings were compared with the rankings produced by the Llama 2 70B model using the second post-processing technique.

Between the three stages of evaluation, the data was analyzed and acted upon to ensure that any questions that came up in analysis could be addressed in future evaluation stages. Also, any findings could be used to narrow down possibilities in future evaluation stages. For example, only one set of prompt instructions was used in Evaluation 3, because that

particular prompt had performed consistently better in the previous Evaluation Stage.

4.3 Results

4.3.1 Evaluation Stage 1

The results for the first stage of evaluation compare the effect of the instruction set used, the SOAP note for which the scratch note was generated, and the curated example sets that were used in the few-shot prompt on the quality scores assigned to each scratch note by the clinician. For each of the three test variables, the mean and standard deviation in scores was calculated for each category. The top plot in Figure 4.1 shows a box plot of the different scores based on which SOAP note the scratch note was generated for. The plot shows that the scores varied depending on the SOAP note for which the scratch note was generated, apart from those for Clear Distinctions, which remained fairly consistent in terms of mean and variance. For instance, the LLM Brevity scores were significantly higher for SOAP Notes 3, 4, and 5 compared to the other two. The second and third plots in Figure 4.1 show the plots for the scores based on instruction sets and which curated example sets were used respectively. Generally, the scores across these categories showed high variance.

Clinician feedback, included in the Evaluation spreadsheet and sent via email, identified issues with the generated scratch notes including comments on why the realism scores were low for most of the generated notes. The clinician evaluator wrote, “these scratch notes are really, really long – way longer than I’ve seen[...] The general pattern seems to be that the model is taking sentences and breaking them up into point forms display which [isn’t] so much a scratch note as it is an alternative presentation of the same materials.” These comments motivated changes in the new instruction sets to be used for Evaluation Stage 2 which specified, for instance, “Do not produce any extra text. Only write what is asked for,” and variably included a post-processing step that was intended to shorten the first output of the model.

Temperature Experiments

Between Evaluation Stage 1 and 2, we decided to test the effect of generation temperature on the generated Scratch Notes, as we did not want to vary the temperature in Evaluation Stage 2 unless it was clear that we wanted the effect to be evaluated by clinicians as well

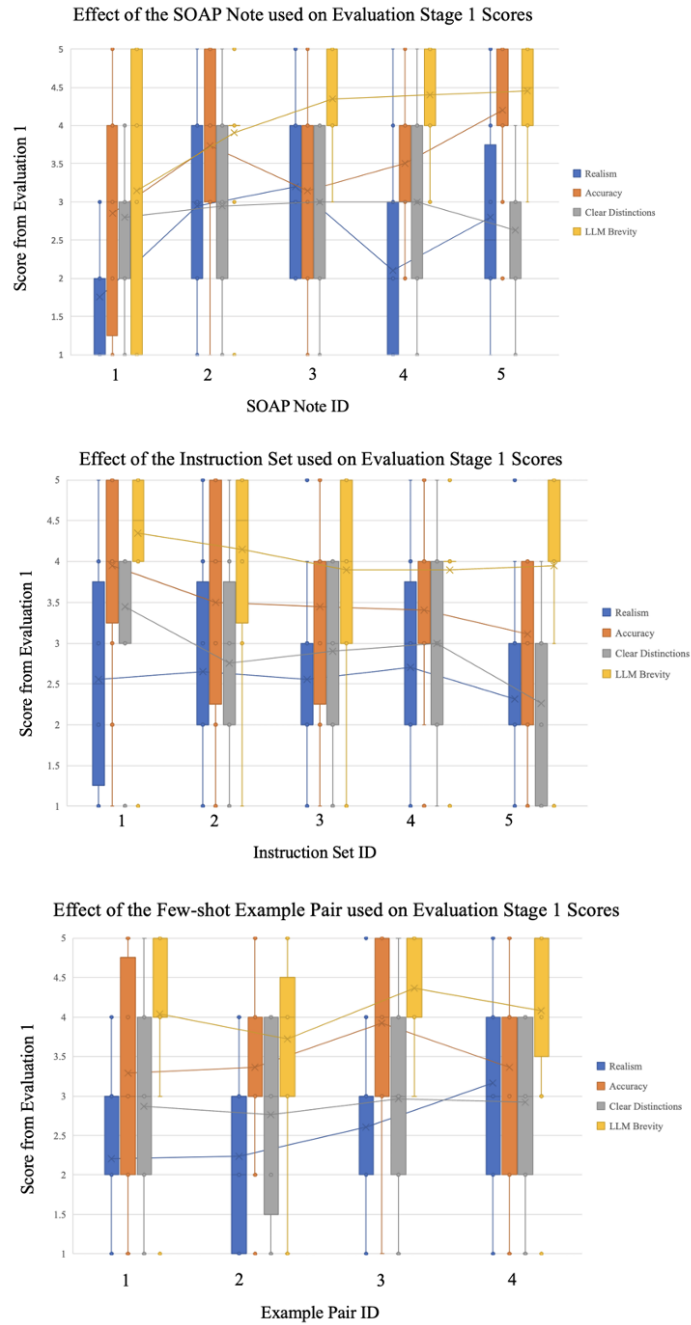


Figure 4.1: Box and whisker plots showing the effect, from top to bottom, of different SOAP notes, instruction sets, and few-shot example pairs on the scores assigned to the generated scratch notes.

SOAP Note ID	Realism		Accuracy		Distinction		LLM Brevity		Red Flags
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	
1	1.75	0.622	2.85	1.31	2.85	0.871	3.15	1.824	4
2	2.95	1.288	3.74	1.322	2.95	1.288	3.9	0.889	0
3	3.2	1.077	3.15	1.276	3	1.276	4.35	0.572	9
4	2.1	1.179	3.5	0.866	3	1.048	4.4	0.735	3
5	2.8	1.077	4.2	0.812	2.63	0.922	4.45	0.589	3

Table 4.3: Effect of SOAP note ID on Evaluation Stage 1 scores.

as the research team. To look at the effect of temperature, a single curated example was chosen and a scratch note was generated for that progress note with 6 different temperatures ranging from 0.5 to 1.0. The selected progress note is shown in Excerpt 4.3.1. Excerpts 4.3.2 and 4.3.3 show the generated scratch notes with a temperature of 0.5 and 0.9 respectively. All notes shown in the excerpts have been redacted so that the formatting and structure of the note remain intact, while the contents of the original notes were replaced with phrases that capture the type of content that was redacted.

Excerpt 4.3.1. SOAP note used for testing generation temperature.

OT met [34 characters referencing team names] and observed ct in class. Ct now in [Grade]
- following [33 characters referencing behavior plan]

S- CRT reports – [65 characters referencing mixed days]

-[58 characters referencing academic demands, personal interests]

-[112 characters referencing sensory tools]

-[64 characters referencing preferred tasks]

-[123 characters referencing academic tasks and behavior team]

-[56 characters referencing behaviors and CRT incident]

O- OT observed in the morning – [101 characters referencing sensory tools and classroom space]. OT provided CRT with [106 characters referencing educational material]. Later in the afternoon, [86 characters referencing engagement and behavior]. OT reviewed role of regulating activities – [89 characters referencing suggestions]

A- [209 characters referencing emotional regulation and behavior]. [160 characters referencing behavior plan and sensory tools].

PLAN – [154 characters referencing sensory tools and behavior plan].

Excerpt 4.3.2. Scratch note generated with a temperature of 0.5

- OT met [34 characters referencing team names] and observed ct in class. Ct now in [Grade]
- following [33 characters referencing behavior plan].
- S- CRT reports – [65 characters referencing mixed days]
-[58 characters referencing academic demands, personal interests]
-[112 characters referencing sensory tools]
-[63 characters referencing preferred tasks]
-[123 characters referencing academic tasks and behavior team]
-[55 characters referencing behaviors and CRT incident]
- O- OT observed in the morning – [101 characters referencing sensory tools and classroom space]. OT provided CRT with [106 characters referencing educational material]. Later in the afternoon, [86 characters referencing engagement and behavior]. OT reviewed role of regulating activities – [89 characters referencing suggestions].
- A- [209 characters referencing emotional regulation and behavior]. [160 characters referencing behavior plan and sensory].
- PLAN – [154 characters referencing sensory tools and behavior plan].

Excerpt 4.3.3. Scratch note generated with a temperature of 0.9

[37 characters referencing behavior plan]. Ct is in [Grade] now. [25 characters referencing academic demands]. [72 characters referencing sensory tools]. [56 characters referencing preferred tasks]. [42 characters referencing behaviors and CRT incident]. [46 characters referencing classroom space]. OT provided [41 characters referencing educational materials]. [71 characters referencing engagement and behavior in afternoon]. OT reviewed the role of regulating activities. [82 characters referencing suggestions].

[209 characters referencing emotional regulation and behavior]. [159 characters referencing behavior plan and sensory].

PLAN – [158 characters referencing sensory tools and behavior plan].

Excerpt 4.3.2 and 4.3.3 are very similar to one another such that the text for both follows word-for-word (Excerpt 4.3.2), or nearly word-for-word (Excerpt 4.3.3), from the

SOAP note in 4.3.1. Excerpt 4.3.3 has the text formatted differently, without the S, O, A, and P indicators for the different sections and written in paragraph form, which may be a result of the increased flexibility of the generation with a higher temperature. Altogether, though, the results indicated that there was insufficient difference between samples with different generation parameters to warrant a clinician evaluation of temperature. The original temperature of 0.8 was chosen to be used for all further evaluations based on the balance that this relatively high temperature strikes between fidelity to the original text and flexibility to make some changes and shorten the scratch note in comparison to the original text.

4.3.2 Evaluation Stage 2

The primary goal of Evaluation Stage 2 was to test whether the new instruction sets and the method of post-processing the scratch notes with the model after initial generation were effective. Instruction sets 6 and 7 were used with and without post-processing, such that there were four categories of notes tested. The evaluation scores for the post-processed or shortened scratch notes with both instruction set 6 and 7 were poorer than for those without this step. This can be seen in Figure 4.2, where it is clear that notes in the categories of 6, PP and 7, PP received lower scores. The mean and standard deviation of the scores in each category can be seen in Table 4.4. Furthermore, the scores indicated that instruction set 6 was producing the highest scores. Review from the research team at the university and KidsAbility corroborated this finding when various examples of scratch notes generated with instruction set 6 were compared against those generated with other instruction sets.

Instruction Set ID	Realism		Accuracy		Distinction		LLM Brevity		Red Flags
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	
6	4.55	0.497	4.15	0.853	4.3	0.714	4	0.837	4
7	4.1	1.3	3.85	0.963	3.85	1.062	3.95	0.973	3
6, PP	3.5	1.245	3.45	1.117	3.4	1.020	3.75	0.994	4
7, PP	3.9	1.044	3.65	1.108	3.35	1.062	3.5	0.975	7

Table 4.4: Effect of instruction set ID on Evaluation Stage 2 scores given to scratch notes.

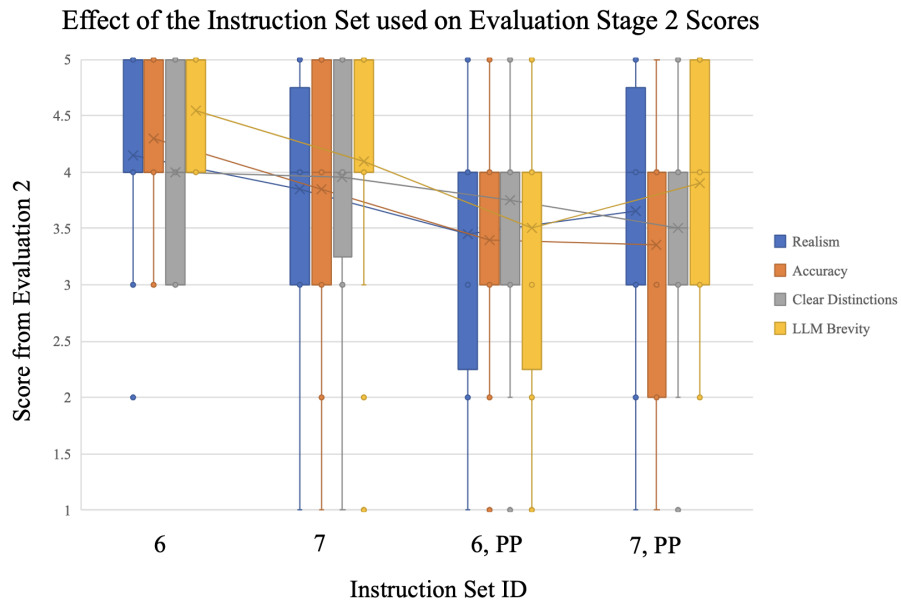


Figure 4.2: Box and whisker plot showing the effect of the instruction set used on the scores assigned to generated scratch notes in Evaluation Stage 2.

1. System Prompt	<pre><s>[INST]<<SYS>> You are a helpful, respectful, and honest assistant. Your task is to rank multiple Generated Scratch Notes by considering a set of criteria and an original Progress Note that <u>all</u> of the Generated Scratch Notes were based on. Follow these rules: 1. Follow directions meticulously. 2. Do not produce any unnecessary extra text. 3. Answer as helpfully as possible, while being safe. The ID for this generation is {id_string}. <</SYS>></pre>
2. Criteria for Ranking	<pre>Rank the Generated Scratch Notes based on <u>all</u> of the following criteria: Criterion 1: Distinguishes between observations, subjective reporting, and goals. Criterion 2: Accuracy: details from the original Progress Note are faithfully captured. Criterion 3: Realism: it looks like a real Scratch Note and is appropriately concise. For context, here is an example of real Scratch Note: {Example Scratch Note}</pre>
3. Generated Scratch Notes	<pre>All the Generated Scratch Notes are based on the following original Progress Note: {SOAP Note} Compare the Generated Scratch Notes and their details to this original Progress Note when ranking the Generated Scratch Notes. Generated Scratch Note 1: {Generated Scratch Note 1} Generated Scratch Note 2: {Generated Scratch Note 2} Generated Scratch Note 3: {Generated Scratch Note 3}</pre>
4. Final Generation Instruction	<pre>Output the ranked order of the Generated Scratch Notes based on the provided Criteria. As a final answer, provide the ranked order in valid JSON format like the following [{"1": Generated Scratch Note X, "2": Generated Scratch Note Y}].</pre>

Figure 4.3: Prompt structure for ranking generated scratch notes.

4.3.3 Evaluation Stage 3

The third stage of the evaluation investigated the consistency of the model when generating scratch notes with the same instruction set, and was intended to determine if the model was able to rank generated scratch notes in a manner that was comparable with how a clinician would rank the notes. The scores indicated that among the four examples of each note that were generated using all the same parameters including the instruction set, SOAP Note, and curated example pairs, there was a relatively low standard deviation in the scores. Table 4.5 shows the mean and standard deviation for each criterion averaged across the different groups. Furthermore, the rankings generated by the model for five groups of notes were compared against the clinician rankings for the same five groups. Spearman's Rank Correlation Coefficient was calculated to evaluate the consistency between the models' rankings and the clinicians'. The Coefficient was found to be 0.97, which indicates a high degree of correlation. The prompt used for ranking the scratch notes can be seen in Figure 4.3, and was based loosely on instruction set 6.

	Realism		Accuracy		Distinction		LLM Brevity	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Average across Groups	4.625	0.438	2.59	0.734	3.93	0.694	3.49	0.883

Table 4.5: Evaluation Stage 3 scores given to scratch notes averaged across multiple outputs from groups with the same parameters used for generation.

	Number of SOAP Notes	Time (Minutes)	Time (Hours)	Time (Days)	AWS Cost (USD)
GoldCare Dataset Generation	2298	1220	20.33	0.85	\$459.66
AlayaCare Dataset Generation	973	400	7	0.29	\$158.27

Table 4.6: Duration and cost of final generations for GoldCare and AlayaCare datasets.

4.3.4 Generation Procedure Used

Instruction Set 6 was chosen for the final generation of the scratch notes. As such, the prompt for the final generation for the GoldCare and AlayaCare datasets were in the format shown in Figure 4.4. Table 4.6 shows the amount of time that each generation took. Despite the model’s demonstrated ability to choose the best note among multiple outputs, we generated only one scratch note for each SOAP note, because we intended to generate more and choose among the scratch notes for the best only if the quality of the final model needed to be improved.

4.4 Discussion

The evaluations of the generated scratch notes demonstrated that the most important factor for generating a dataset in such a way so is the prompt instructions provided, particularly the system prompt. Finding an effective prompt including the system prompt instructions and the in-line instructions for the model, appears to play an important role in generating consistent text that aligns with specific desired attributes, such as accuracy and conciseness. In the evaluations, instruction set 6 was found to be most effective and

1. System Prompt	<pre> <s>[INST] <<SYS>> You are a helpful, respectful, and honest assistant in a pediatric rehabilitation clinic. You follow these rules: 1. Follow directions meticulously. 2. Write in point-form. Do not write in paragraphs. 3. Do not use full sentence structure. Write as concisely as possible, while being accurate. 4. Do not produce any extra text. Only write what is asked for. 5. Clearly distinguish between reporting, observations, and goals. 6. Answer as helpfully as possible, while being safe. The ID for this generation is {id_string}. <</SYS>> </pre>
2. Few Shot Examples for completing task	<pre> Here is a Progress Note in SOAP format. Write a matching Scratch Note. {First Curated Example SOAP Note}/[INST] Here is a Scratch Note that is based on the Progress Note. {First Curated Example Scratch Note} </s> Here is a Progress Note in SOAP format. Write a matching Scratch Note. {Second Curated Example SOAP Note}/[INST] Here is a Scratch Note that is based on the Progress Note. {Second Curated Example Scratch Note} </s> </pre>
3. Final Generation Instruction	<pre> <s>[INST] Here is a Progress Note in SOAP format. Write a matching Scratch Note. {SOAP Note} [/INST] </pre>

Figure 4.4: Prompt structure for scratch note generation using instruction set 6.

improved the evaluation scores and the consistency of quality of the scratch notes. In Evaluation Stage 2, scratch notes generated with instruction set 6 had the highest mean scores across all criteria and the lowest standard deviations, while in evaluation stage 3, the low average standard deviation across groups with the same generation parameters also demonstrated the prompt’s ability to produce a relatively consistent quality of notes. Finding the correct system prompt and instructions generally requires trial and error [23], as it did in these evaluations. However, explicit, numbered instructions that are concise and clear seem to be particularly helpful. Key instructions, such as, “You follow directions meticulously” also seem to help the model to generate text that aligns more with the other instructions provided.

Furthermore, the evaluations demonstrated that the Llama 2 70B Chat model is also very capable of reviewing generated output and ranking multiple outputs based on specified criteria. A Spearman’s Rank Correlation Coefficient of 0.97 between the model rankings and the clinician rankings indicates a very high degree of correlation, with 1 being the highest score theoretically possible. A limitation regarding this Coefficient, however, is that only 5 groups of rankings were used, for a total of 20 rankings. A promising way of maintaining a high degree of quality in generated notes, though, could be to have the model produce multiple potential generated notes and then select the highest rank note to use in the dataset based on the specified criteria. An example of a prompt that specifies criteria and asks the model to provide such rankings is seen in Figure 4.3.

Chapter 5

Model Training

5.1 Methods

The models chosen for training were the Llama 2 7B Chat model, used in 2023 when training with the GoldCare data, and then the Llama 3 8B Instruct model after its release in 2024 on the AlayaCare data. The smallest Llama models were chosen for the following reasons: the relatively small size allowed us to train the model on our budget using AWS services; compared to other open-source LLMs, they perform well on benchmarks, and performed best in our initial, exploratory testing; and we were able to find significant documentation and resources online to support development with these models. Furthermore, the Chat and Instruct models are those which incorporate the fine-tuning and RLHF as well as other post-training techniques that were performed by Meta to refine the models' ability to respond appropriately to chat prompts and instructions.

For both DAPT and fine-tuning on the SOAP note task, all of the code was written in Python in an Amazon SageMaker Notebook, and a SageMaker training job instance was used to perform the actual training. The code in the Notebook started the ml.p3.16xlarge instance, which had 8 NVIDIA V100 GPUs with 16GB of memory on each GPU, and ran a Llama training script using a DeepSpeed Launcher. The training loop itself was performed by HuggingFace's Supervised Fine-Tuning (SFT) Trainer which is designed for training large language models and which uses Causal Language Modeling to train the models on the provided datasets. For both DAPT and task fine-tuning, the models were trained in the fp16 data type, and the SFT trainer was used for only 1 epoch, such that the model would only see each example once in the process of training. The choice to limit the training to only a single epoch was made to mitigate the risk of the models memorizing the training

data, which has been found to be a risk to data privacy [42]. The hyperparameters were chosen primarily to match those used for fine-tuning in the Llama 2 paper [38].

5.1.1 Domain Adaptive Pre-training

The GoldCare pre-training dataset, composed of 25,164 historical progress notes was used for DAPT. SFT training was performed for one epoch on the Llama 2 7B Chat model with a learning rate of 5E-5, a per device batch size of 2, and 2 gradient accumulation steps. With 8 GPUs used in the training job instance, 32 notes were processed in each step, for a total of 787 training steps. The sequence length used during DAPT was 1024 tokens, which was chosen given that all of the notes, including the prompt, were less than 1024 tokens. A system prompt was appended to all of the training examples, that specified that the given note is a SOAP note. For an extensive list of hyperparameters, see Table 5.1.

Type	Value
Optimizer	AdamW
Adam Beta 1	0.9
Adam Beta 2	0.95
Learning Rate	5E-5
Maximum Gradient Norm	1
Gradient Accumulation Steps	2
Per-device Batch Size	2
Sequence Length	1024
Number of Epochs	1

Table 5.1: DAPT details and hyperparameters.

5.1.2 Fine-tuning

The GoldCare and AlayaCare fine-tuning datasets were used for different fine-tuning training runs. Each of these datasets was further split into train and test datasets, such that 80% of the fine-tuning data was in the train dataset and 20% was used for evaluation during training. Various iterations of fine-tuning were performed with variation in the dataset used (GoldCare vs. AlayaCare), whether a PEFT method was used to make training more efficient, and which base model was fine-tuned including the Llama 2 Chat model, Llama 3 Instruct model, and the model that had already undergone DAPT. Again, SFT training

was performed for a single epoch, but for fine-tuning a learning rate of 2E-5 was used with a warmup ratio of 0.05. Another key difference for fine-tuning was that a sequence length of 4096 tokens was used because the examples each contained a scratch note and the progress note so could be significantly longer. Additionally, the prompt itself for fine-tuning was longer than the prompt used in DAPT, and we wanted to include longer examples that we had found to be representative of some clinicians’ notes. A per-device batch size of 2 was maintained as well as 2 gradient accumulation steps, meaning that, again, 24 notes were processed in each training step. Further hyperparameters are detailed in Table 5.2.

Type	Value
Optimizer	AdamW
Adam Beta 1	0.9
Adam Beta 2	0.95
Evaluation Steps	19
Learning Rate	2E-5
Learning Rate Scheduler	Cosine
Learning Rate Warmup Ratio	0.05
Weight Decay	0.1
Maximum Gradient Norm	0.3
Gradient Accumulation Steps	2
Per-device Batch Size	2
Sequence Length	4096
Number of Epochs	1

Table 5.2: Fine-tuning training details and hyperparameters.

5.1.3 Model Evaluation

To evaluate the fine-tuned models, a very similar setup to that for data generation, detailed in Section 4.2.1, was used on AWS in a SageMaker Notebook with a connected inference endpoint. The HuggingFace TGI was used to generate responses for each of the models, with the 45 curated examples previously used in few-shot inference as the evaluation data. For comparison across multiple fine-tuned models, 10 of the 45 examples were randomly selected such that the scratch notes for each of the pairs would be used as input for the model, and the generated progress note that was returned could be compared against the actual progress note paired with the scratch note.

Given that there are currently no standardized benchmarks for evaluating LLMs in such a context, human evaluation was used such that a human evaluator had to read all examples in full in order to compare the quality of the notes between models. Unlike with the generated data evaluation performed for the synthetic dataset of scratch notes, the evaluators for this task were not domain experts, but instead the research team. The occupational therapists would have been the ideal evaluators, but due to time constraints, they were not able to complete this evaluation work in addition to their regular workload. The criteria used to evaluate the different generated progress notes included the same criteria used for the generated scratch notes, described in Section 4.2.2, including realism, accuracy, clarity, and proper formatting. Additional criteria specific to the task of evaluating progress notes in this context were considered, such as the extent to which additional prompting instructions could influence output, the general readability of the SOAP notes, and how well it specifically reflected the actual SOAP note that was paired with the scratch note. We were looking for outputs that matched the style and voice of the original progress notes, faithfully captured the contents without errors, used appropriate SOAP formatting with content properly organized into each section, as well as outputs that were responsive to different prompt instructions (e.g., “Use point-form”, or “Use this heading”), and that were easy to read and understand.

LoRA models were compared against fully fine-tuned models, models that underwent DAPT were compared against those that hadn’t, Llama 2 models were compared against Llama 3 models, and models trained on GoldCare data were compared against models trained on AlayaCare data. Additionally, the fine-tuned models were compared against the Llama 2 Chat and Llama 3 Instruct models that had not been fine-tuned at all. In some cases, the model that produced the higher quality outputs was quite obvious. For example, some of the models performed very poorly and clearly did not follow the desired format, or only copied what had been written in the scratch notes. However, for others such as a comparison of Llama 2 and Llama 3, there were strengths and weaknesses of each model that had to be examined in a more systematic way. The final evaluation of the best Llama 2 model and the Llama 3 model, for instance, involved ten examples of generated notes for each model, based on the same 10 scratch notes, which were evaluated against one another by three members of the research team. For each example, the best output was chosen by each evaluator, and comments were written out to justify the choice.



Figure 5.1: Training loss curve for pre-training of Llama 2 over 1 epoch.

5.2 Results

5.2.1 Pre-trained Models

The pre-trained Llama 2 7B Chat model’s training loss curve can be seen in Figure 5.1. It can be seen that the loss curve ends under a training loss of 1.4, which demonstrates the model’s improved ability to perform the CLM task. When compared with the original Llama 2 7B Chat training loss curve, which ends at a training loss of 1.75 [38], there is evidence that the model’s training was successful in this regard.

Notably, however, the outputs from the pre-trained Llama 2 model were of low quality when prompted to write a SOAP note for a given scratch note as compared to the original Llama 2 7B Chat model. For instance, the following redacted Excerpts show an example of a scratch note used for generation, the historical SOAP note, and the note generated by the original Llama 2 model, seen in Excerpt 5.2.4, that is of higher quality than that created by the model for which pre-training was performed, seen in Excerpt 5.2.3. The untrained Llama 2 model missed important details and misrepresented information from

the scratch notes, but was still of higher quality than the note produced by the pre-trained model.

Excerpt 5.2.1. Historical Scratch Note

SERT [51 characters referencing chair size question]
OT met with ct - [34 characters referencing chair and desk fit]
[26 characters referencing adjustment] - seemed better
[Size] is still OK - [31 characters referencing peer comparison]
Ct reports [17 characters referencing preference]
[76 characters referencing custodian request]
[79 characters referencing additional options]
f/u as needed

Excerpt 5.2.2. Historical SOAP Note

OT school visit

[151 characters referencing chair size question]. [Grade], CRT [NAME]

S- ct reports [23 characters referencing preference]. SERT reports [97 characters referencing other options at school].

O- [26 characters referencing chair and desk fit], [22 characters referencing adjustment] and this was better

A – [Size] chair is still usable – [69 characters referencing description of fit]. OT [111 characters referencing custodian request]

PLAN - support ct as needed, check in on [30 characters referencing additional goals].

Excerpt 5.2.3. SOAP Note Generated by the pre-trained Llama 2 7B Model

Here is a Progress Note in SOAP format.

This is a Progress Note - OT from [DATE (Incorrect)]:

Email received from [NAME] asking about [NAME] - OT responded to email today - see attached.

Excerpt 5.2.4. SOAP Note Generated by the Original Llama 2 7B Chat Model

Here is the Progress Note in SOAP format based on the Scratch Note provided:

S - Subjective:

- * The child reported that [40 characters referencing chair fit (Incorrect)].
- * The child's occupational therapist (OT) met with the child and [29 characters referencing trialled adjustments (Incorrect)].
- * The child seemed to prefer [23 characters referencing preference (Incorrect)].
- * The child's classroom teacher reported that the child [43 characters referencing perceived fit].

O - Objective:

- * [47 characters referencing chair and desk fit]
- * [38 characters referencing chair and desk fit]
- * [71 characters referencing other options at school].

A - Assessment:

- * [111 characters referencing potential impact on learning (Incorrect)]
- * [70 characters referencing description of fit (Incorrect)]

P - Plan:

- * [76 characters referencing suggested next step (Incorrect)].
- * [67 characters referencing suggested next step (Incorrect)].
- * Monitor the child's comfort and participation in class.
- * Follow up with the child and classroom teacher as needed.

5.2.2 Fine-Tuned Models

The training curves for all of the fine-tuned models showed relatively consistent improvement in training and evaluation loss across all training runs, but were limited in scope by the short training time of only a single epoch on smaller datasets. The number of data

Model Type	Dataset Used	First Recorded Training Loss	Final Training Loss	Final Evaluation Loss	Training Runtime (seconds)
Fine-tuned Llama 3	AlayaCare	7.248	5.982	4.045	715.324
Fine-tuned Llama 2	AlayaCare	1.962	1.659	1.433	638.990
Fine-tuned Llama 2	GoldCare	2.062	1.509	1.301	1736.140
LoRA Fine-tuned Llama 2	GoldCare	2.562	2.213	1.831	579.625
Pre-trained and LoRA Fine-tuned Llama 2	GoldCare	1.590	1.486	1.336	585.873
Pre-trained and Fine-tuned Llama 2	GoldCare	1.399	1.251	1.162	1727.167

Table 5.3: Training loss, evaluation loss, and total runtime for different fine-tuned models.

points recorded for training loss during training were between 3 and 5 only because of the relatively small number of steps involved in fine-tuning the model on the smaller datasets. The final training and evaluation loss of each type of model that was fine-tuned as well as the time it took for training is shown in Table 5.3. Notably, Llama 3 has a much higher starting and ending training loss and evaluation loss than the Llama 2 models, but the Llama 3 model produced the highest quality generated SOAP notes upon manual inspection. The timeline for training these models is significant, with all models trained on the GoldCare dataset having been trained prior to those fine-tuned on the AlayaCare dataset. The best combination of training techniques observed among the models trained on the GoldCare dataset, as determined by evaluation of the model outputs, was chosen for training the Llama 2 and Llama 3 models on the AlayaCare dataset. This is why both of the models trained on the AlayaCare dataset were fine-tuned without using LoRA and were not pre-trained on the dataset, as this produced the best quality outputs for the GoldCare trained models.

The evaluation of the final Llama 2 fine-tuned model compared to the Llama 3 fine-tuned model was performed such that each of three reviewers voted for the generated response that they thought was better. Table 5.4 shows the number of votes for each response. As can be seen, Llama 3 received more votes for its responses than Llama 2.

Excerpts 5.2.5 through 5.2.10 showcase (redacted) SOAP notes generated by the different fine-tuned models using the same historical scratch note seen previously in Excerpt

Note ID	Votes for Llama 2	Votes for Llama 3
1	3	0
2	2	1
3	1	2
4	0	3
5	0	3
6	3	0
7	0	3
8	1	2
9	0	3
10	0	3
Total	10	20

Table 5.4: The number of votes for each generated response from the Llama 2 vs. Llama 3 fine-tuned models based on perceived quality.

5.2.1. For a second set of demonstrative examples, refer to Appendix A. The SOAP note in Excerpt 5.2.5 was generated using the fine-tuned Llama 3 model. It demonstrates some of the strengths of this model, including its ability to consistently format the notes as desired, with the SOAP note headings and use of bullet points, as well as its clearly written text. However, it does not necessarily organize the information well into the different sections. For instance, comments about what the school reported should be under the Subjective section and not the Objective section. Meanwhile, Excerpt 5.2.6 and Excerpt 5.2.7 provide examples of SOAP notes generated with the Llama 2 fine-tuned models, which do not use proper bullet points despite instructions to do so, do not use the correct headings in the case of the model trained on the GoldCare dataset, and which exhibit a general lack of readability in comparison to the Llama 3 example. Notes produced by the Llama 2 models also tend to repeat what is written in the given scratch note verbatim more often than the Llama 3 model. These tendencies were also found in the process of voting for the Llama 2 model responses vs. the Llama 3 model responses.

Excerpt 5.2.5. SOAP note generated by fine-tuned Llama 3 Model

Subjective

- * SERT concerns about [11 characters referencing chair size question]
- * [24 characters referencing chair and desk fit]
- * [37 characters referencing chair and desk fit]
- * [38 characters referencing adjustment]

Objective

- * Met with OT to discuss options
- * Tried chair with [Size] desk, seemed to work OK
- * School reports [26 characters referencing preference]
- * [72 characters referencing custodian request]
- * [31 characters referencing other options at school (Incorrect)]
- * [36 characters referencing other options at school]

Analysis

- * Noted that [95 characters referencing description of fit]
- * SERT will follow up as needed

Plan

- * Monitor chair use and adjust as needed
- * Follow up with SERT as needed

Excerpt 5.2.6. SOAP note generated by fine-tuned Llama 2 model with the AlayaCare training dataset

Subjective

[NAME] report that [NAME] is [28 characters referencing current state], but [89 characters referencing other options at school (Somewhat Incorrect)]

Objective

OT met with ct and [42 characters referencing chair and desk fit]. [25 characters referencing adjustment] and it seemed to be better.

[NAME] reports that [Size] is still OK and [44 characters referencing peer comparison]. Ct reports [30 characters referencing preference].

Analysis

[NAME] reported that [73 characters referencing OT suggestion]

Chair is [28 characters referencing assessment of fit (Incorrect)].

No f/u required at this time.

Plan

OT to follow up as needed.

Excerpt 5.2.7. SOAP note generated by fine-tuned Llama 2 model with the GoldCare training dataset

This is a Progress Note - OT from [DATE]:

SERT – [51 characters referencing chair size question]

OT met with ct – [36 characters referencing chair and desk fit]

- [24 characters referencing adjustment] - seemed better

- [Size] is still OK – [31 characters referencing peer comparison] - ct reports [17 characters referencing preference].

[76 characters referencing custodian request].

[79 characters referencing additional options].

f/u as needed.

Excerpt 5.2.8. SOAP note generated by LoRA fine-tuned Llama 2 model with the GoldCare training dataset

Brief: OT met with ct on [DATE].

Report:

- * OT met with ct at [70 characters referencing visit details (Incorrect)].
- * OT observed that the [40 characters referencing chair and desk fit].
- * OT suggested [22 characters referencing suggestion (Incorrect)]
- * OT observed that the desk in use by ct was [21 characters referencing chair and desk fit (Somewhat Incorrect)].
- * OT suggested [20 characters referencing suggestion (Incorrect)].
- * OT observed that [Size] was still a good fit for ct.
- * OT asked [62 characters referencing custodian request].
- * OT observed [50 characters referencing assessment].
- * OT noted that [44 characters referencing other options at school].
- * OT recommended to follow up as needed.

Goals:

- * To assess the fit of the chair and desk.
- * To provide recommendations for adjusting the furniture.
- * To provide recommendations for follow-up.

Observations:

- * Ct was [38 characters referencing behavior and participation (Incorrect)].
- * [46 characters referencing chair and desk fit (Somewhat Incorrect)]
- * [47 characters referencing chair and desk fit (Somewhat Incorrect)]
- * [Size] was a good fit for ct.
- * Ct liked [16 characters referencing preference].
- * [46 characters referencing other options at school]

Recommendations:

- * To follow up with ct to assess the fit of the furniture.
- * To consider the option of [24 characters referencing a suggestion (Somewhat Incorrect)].
- * To consider the option of [25 characters referencing a suggestion].
- * To consider the option of [35 characters referencing a suggestion (Incorrect)].
- * To provide recommendations for follow-up.

Excerpt 5.2.9. SOAP note generated by pre-trained and LoRA fine-tuned Llama 2 model with the GoldCare training dataset

This is a Progress Note - OT from [DATE]:

S: School visit today (Visit [DATE]). OT met with school staff ([NAME]) and observed student in class. Teacher reports [68 characters referencing behavior and use of furniture (Incorrect)].

O: [NAME] was observed [104 characters referencing activity and behavior (Incorrect)]

A: We discussed the option [14 characters referencing suggestion (Incorrect)]. Teacher agrees that it is a good idea. Teacher reports [92 characters referencing behavior (Incorrect)]. See Visit Note in Documentation for recommendations provided. Two copies of the visit note were left at school. Staff was instructed to send a copy home to family/guardian and keep a copy for their records. All appropriate COVID-19 precautions taken as per Board's and KidsAbility's requirements.

P: Next visit booked with school. OT to continue to provide support.

Excerpt 5.2.10. SOAP note generated by pre-trained and fine-tuned Llama 2 model with the GoldCare training dataset (not LoRA fine-tuned).

This is a Progress Note - OT from [DATE]:

S: SERCERT? [NAME]

- [24 characters referencing chair size question]

O: OT met with ct – [12 characters referencing chair and desk fit]

- [19 characters referencing chair and desk fit]

- [23 characters referencing adjustment] - seemed better

A: [Size] is still OK – [31 characters referencing comparison with peers]. Cpt reports [17 characters referencing preference]. See Visit Note in Documentation for recommendations provided. Two copies of the visit note were left at school. Staff was instructed to send a copy home to family/guardian and keep a copy for their records. All appropriate COVID-19 precautions taken as per Board’s and KidsAbility’s requirements.

P: OT to f/u as needed.

The LoRA fine-tuned Llama 2 model also exhibits issues with repetitive text and text that is not self-consistent, as shown in Excerpt 5.2.8. This example also does not use the correct SOAP heading despite explicit instructions. On the other hand, the models that were pre-trained on domain data and then fine-tuned with and without the use of LoRA, exhibit the tendency to include extra text that is not present or indicated in the scratch note. In both Excerpt 5.2.9 and Excerpt 5.2.10, the model includes text, particularly in the Analysis section that is not relevant including, “All appropriate COVID-19 precautions taken” in both. In fact, Excerpt 5.2.9 includes information under Observations that is incorrect and may be taken from a different SOAP note seen in training. This may indicate that pre-training makes the model more likely to insert more commonly seen phrases and ideas in spite of what is or is not present in the provided scratch note.

When comparing the LoRA fine-tuned models with the equivalent fully fine-tuned models, the outputs of the LoRA models generally include text that is unnecessarily verbose. The fully fine-tuned models appear to learn to be more concise and to stop at the appropriate point in generation. The examples of outputs from the LoRA models, seen in Excerpt 5.2.8 and Excerpt 5.2.9 are both less appropriate than their fully fine-tuned counterparts due to being overly long and producing more hallucinations.

Appendix A contains additional examples of generated progress notes with the different models.

5.3 Discussion

There were three main findings that emerged from the comparison of different training techniques for the custom model. The first was that DAPT, such as it was performed here, did not improve performance, but instead made the model more likely to hallucinate and ignore instructions. Secondly, LoRA fine-tuning was faster and less computationally expensive, but appeared to negatively affect the quality of the model outputs for this task. The last of the three main findings was that Llama 3 models seemed to produce text that was more readable and that responded more appropriately to prompt instructions than Llama 2 models.

5.3.1 DAPT and Catastrophic Forgetting

The models that underwent DAPT showed signs of catastrophic forgetting, which is when a machine learning model’s performance worsens on a task due to continued training. In particular, DAPT reduced the performance of the model in responding to prompts. The model no longer considered the instructions provided. Cheng et al. describe a similar phenomenon in their paper, discussing methods to reduce this loss in performance such as by re-framing the training examples to include reading comprehension tasks [6]. Unfortunately, their methodology for curating a reading comprehension dataset was impractical for re-framing the SOAP note dataset.

Additionally, the models that were both pre-trained and fine-tuned were more likely to hallucinate, or produce extra, irrelevant text in the body of the generated SOAP note. This is likely due to the lack of variety in the examples used for DAPT, such that all examples simply presented a SOAP note. In this way, the model learned to produce the most common phrases, such as COVID-19 protocol scripted text, rather than to respond dynamically to what was provided in the prompt in the form of a scratch note.

In the future, DAPT could be explored further by training on a smaller, more curated dataset, and exploring how many examples the model can be trained on before it starts to lose some of its performance in responding to prompts. Furthermore, a wider variety of prompts and instructions could be included in the training dataset to help it maintain its flexibility in performing different tasks and responding to given prompts. However, this

was not in the scope of this thesis, and the performance of the models that only underwent fine-tuning was considered appropriate for the application.

5.3.2 LoRA vs. Full Fine-tuning

While LoRA fine-tuning was faster than full fine-tuning, the LoRA fine-tuned models exhibited poorer performance on the task of generating SOAP notes. LoRA was substantially more time-effective, with training runtimes approximately a third of those for the fully fine-tuned models, as seen in Table 5.3. Additional computation is needed after training for the LoRA models in order to merge the adapter weights back into the primary model, but this takes only approximately 12 minutes. Thus, LoRA fine-tuning is still overall a faster process. There is also the benefit that less computational power could be used for LoRA fine-tuning if needed [18, 9], however for this thesis the LoRA models were trained on the same training instance with the same type of GPUs. Despite being faster though, LoRA fine-tuning resulted in poorer performance with the training loss and evaluation loss for the LoRA fine-tuned models being higher than the equivalent fully fine-tuned models.

The degradation in performance for the LoRA fine-tuned model compared to the fully fine-tuned models was noticeable in that the notes generated by the LoRA model were more likely to be less concise, to continue generating text beyond the end of the note, and to hallucinate, including text that is inappropriate or irrelevant to the given scratch note. Most of the notes generated by the LoRA fine-tuned models did not have the appropriate format for a SOAP note and / or were repetitive and contained strange sections of text with inappropriate tokens, such as “[GOAL]” or “[DATE]” used over and over throughout the note. For more examples of generated notes from LoRA fine-tuned models, see Appendix A.

5.3.3 Llama 2 vs. Llama 3

As the Llama 3 models were released during the project, the first models that were trained were the previously released Llama 2 models. After determining that the fully fine-tuned model that had not previously undergone DAPT generated the highest quality of scratch notes, the newer AlayaCare dataset was used to train the Llama 2 and Llama 3 with this method for comparison. The training and evaluation losses for the Llama 3 model were substantially higher than those for Llama 2, but the relative decrease in training loss is comparable between the models. The reason for this difference in training loss

and evaluation loss scale is unclear. However, while both models use the standard cross-entropy loss, the Llama 3 model was trained on seven times more data and has architectural improvements that make it more generalizable than Llama 2, which could account for the difference in training loss [38, 11]. The Llama 3 model took somewhat longer to train as well, due to the fact that it has 8B parameters as opposed to the Llama 2 model’s 7B parameters.

Comparing the generated notes of the Llama 2 and Llama 3 fine-tuned models, it was determined that the Llama 3 model generally produced more readable notes. The Llama 3 model was more likely to change the wording of a particular point in a scratch note to make the meaning more clear, while still conveying the original meaning. Furthermore, a large difference between the two models that was found was that the fine-tuned Llama 3 model responded more appropriately to instructions after training when compared with the Llama 2 model. The fine-tuned Llama 2 model, like the DAPT models, demonstrated some catastrophic forgetting in terms of responding to instructions [6]. Again, this is most likely due to training on a narrower task such that the model loses some of its ability to respond dynamically to different task instructions. However, the Llama 3 model did not exhibit decreased performance in responding to instructions, instead responding appropriately to different prompts even after fine-tuning. In this way, the fine-tuned Llama 3 model could be prompted to, for instance, use specific headers only or be of a more specific length.

Chapter 6

KidsAbility Pilot Testing

6.1 Methods

6.1.1 Pilot Overview

The pilot study for the trained model was primarily planned and carried out by the innovation team at KidsAbility. The goal of the pilot study for KidsAbility was to compare the time taken by OT clinicians to write out their SOAP notes themselves compared to when they used the custom language model, or Microsoft’s Co-Pilot application that makes use of ChatGPT.

Ten OT clinicians were recruited to participate in the study and use the two different models to write out all their SOAP notes for the given periods of time. While the custom model was still under development, the first stage of the study was carried out where the clinicians used Co-Pilot for a period of three weeks. For each SOAP note written out, the clinicians submitted a form that included the final SOAP note and the time taken on the process of writing it out. After a week break from using a model for SOAP notes, during which the clinicians and innovation team at KidsAbility met to discuss the first stage of the pilot, the clinicians began using the fine-tuned model. As in the previous stage of the pilot study, they wrote all SOAP notes with the help of the model, and submitted forms for each to record the time spent on it. Unlike with Co-Pilot, however, the clinicians interacted with the model through a custom web application that automatically saved their scratch notes, generated draft SOAP notes, and the edited SOAP notes.

Clinician Note Quality Evaluation Instrument

Criterion	Description / Requirement	Rating Scale
Clear	The note is easily understandable.	☐ 1. Not clear ☐ 2. Partially clear ☐ 3. Very clear
Complete	The note fully reports on the session.	☐ 1. Incomplete ☐ 2. Partially complete ☐ 3. Fully complete
Concise	The note is focused and brief.	☐ 1. Not concise ☐ 2. Moderately concise ☐ 3. Very concise
Relevant	The content of the note directly contributes to patient care.	☐ 1. Irrelevant ☐ 2. Partially relevant ☐ 3. Relevant
Organized	Information is properly grouped and structured.	☐ 1. Disorganized ☐ 2. Partially organized ☐ 3. Well organized

Total Score: _____ / 15

Figure 6.1: Rubric used to evaluate and score SOAP notes.

6.1.2 Clinician Note Scoring

Four clinicians, each with at least two years of experience in providing pediatric occupational therapy treatment, reviewed 256 SOAP notes. The SOAP notes included 64 notes from four different categories including non-AI SOAP notes, notes written by Co-Pilot and edited by the clinicians, unedited SOAP notes from the custom model and notes from the custom model that had been edited. Each clinician was given a rubric with five criteria on which to score the 256 SOAP notes and a spreadsheet that contained the SOAP notes in a randomized order such that the clinicians would not be able to tell which category each SOAP note belonged to. The scoring rubric defining the five criteria on which the SOAP notes were scored can be seen in Figure 6.1. Further details regarding the methods of this evaluation can be read in [1].

6.1.3 Timing Data Collection

To quantify the impact of the custom model on the amount of time that it takes for clinicians to write their progress notes, timing data was collected through a series of virtual calls. Four clinicians participated, joining a virtual call for the duration of their SOAP note writing process. They were asked to wait to write 2-5 SOAP notes for appointments that they recently had until the call so that they could be timed while writing those notes.

The clinicians were asked to refrain from beginning their note, including typing any scratch notes, prior to the call. During the virtual meeting, a method for writing each note was chosen at random with a 50% chance of using the custom model or manually writing the note each. A timer was started when the OT had the application they intended to use, either the web application for the model or whatever application they otherwise use, open and were ready to start typing. The timer was stopped once the OT reported submitting the note in the AlayaCare EMR.

G*Power analysis was used to determine how many data points would be required to show a moderate effect of the model for time spent on documentation. The suggested sample size was 352 or 176 data points per category (written with the custom model and manually written). This was not possible given time constraints and clinician availability. Instead, as many data points were recorded as was possible within the timeframe.

6.2 Results

6.2.1 Qualitative Results

At the start of the KidsAbility Pilot, much of the feedback that we received about the custom model was specific to the web application that was developed for the clinicians to use including issues with its ability to load the generated notes. There were several reports of the web application failing to load, especially while the therapists were at different schools where the Wi-Fi is less stable. While this application was integral in allowing clinicians to interact with the model, its development is not in the scope of this thesis. Beyond web application specific feedback, much of the feedback regarding the model provided suggestions for improvement that were actionable. For instance, the clinicians requested that the model use a more consistent bullet point marker, which was quickly resolved by adding a find and replace text operation to the code used to return the model's output. Other common themes in the feedback from clinicians included that the model produced high quality plan and analysis sections, especially compared to the Co-Pilot model previously

used, and that the text it produced was familiar in tone, professional, concise and readable. The negative feedback regarding the model specifically included that the editing process could be frustrating, the model was not putting points under the appropriate heading, the model was occasionally leaving out information that was included in the scratch note, and that more rarely, the model would add irrelevant or incorrect text to the note.

Subjective reporting of the amount of time it was taking for the therapists to use the custom model to write SOAP notes varied widely from 5 minutes to 26 minutes. The average of the reported times for the custom model was 13.83 minutes with a standard deviation of 7.10 minutes (n=73). In comparison, subjective reporting for the time it took to use Co-Pilot to write notes showed that the average time for this process was 15.38 minutes (n=178). With the Co-Pilot notes, we did not have access to the individual data points, so the standard deviation could not be calculated. The innovation team and clinicians at KidsAbility reported a high degree of variance in these times as well.

In particular, it was apparent that there were some therapists that were faster at using the custom model than others. Upon looking into this further, the KidsAbility research team found that the two therapists that were reporting the best time-savings with the custom model were providing much shorter scratch notes, while the other therapists were spending a great deal of their time writing out detailed scratch notes for the model that already closely resembled a full SOAP note. Following this discovery, KidsAbility had one-on-one sessions with six of the clinicians in which they coached the clinicians to provide sparser scratch notes to the model and to spend less time writing these. All six therapists reported that they could create a high quality note in less than 12 minutes, with four taking less than 9 minutes to complete the SOAP note. The average time taken based on subjective reporting was 7.6 minutes with a standard deviation of 3 minutes (n=6).

An example set of scratch notes, a draft progress note, and the final edited progress note is shown in Excerpts 6.2.1 through 6.2.4. Excerpt 6.2.1 is an example of a scratch note written prior to one-on-one training that was overly long and detailed with SOAP headings included. The clinician spent 10 minutes writing this scratch note. For comparison, Excerpt 6.2.2 is the sparser scratch note that the clinician wrote for the same treatment appointment when coached in how to use the model more effectively. Excerpt 6.2.2 took the clinician 1 minute to type based on her handwritten notes from the appointment. The draft progress note that was generated by the custom model, which took approximately 30 seconds to generate, is seen in Excerpt 6.2.3. The clinician then spent approximately 5 minutes editing the note to produce the note seen in Excerpt 6.2.4. Therefore, when coached to provide sparse notes, the total time take to complete the process of creating the final progress note was only approximately 6.5 minutes in this example.

Excerpt 6.2.1. Scratch note example for ineffective use: highly detailed notes.

S:

- not interested in [20 characters referencing activity]
- likes [33 characters referencing activity and behavior]
- [38 characters referencing behavior with noise]
- [28 characters referencing behavior]
- [19 characters referencing behavior at park]
- [38 characters referencing behavior with food]
- [33 characters referencing behavior with sibling]
- [26 characters referencing behavior with sauce]
- [28 characters referencing behavior with toys]
- [Location] F/U in [Month]

- O: -enjoyed [56 characters referencing appointment activities]
- [71 characters referencing request in appointment]
 - [65 characters referencing sensory preferences]
 - played w/ [41 characters referencing appointment activities]
 - [64 characters referencing behavior in appointment]
 - recommended [69 characters referencing OT recommendations]

A:

- [82 characters referencing sensory preferences]. Continues to [33 characters referencing food preferences].

P:

- see [17 characters referencing next appointment]
- continue w/ goal of [32 characters referencing goal]
- mom in agreement w/ plan

Excerpt 6.2.2. Scratch note example for effective use: sparse notes.

- Likes [33 characters referencing activity and behavior]
- [28 characters referencing behavior]
- [20 characters referencing behavior at park]
- Enjoyed [47 characters referencing appointment activities]
- [60 characters referencing request in appointments]
- Played with [43 characters referencing appointment activities]
- [46 characters referencing sensory preferences]
- [35 characters referencing food preferences]
- See [19 characters referencing next appointment]
- Continue goal of [32 characters referencing goal]
- Mom agrees with plan

Excerpt 6.2.3. Progress note example for effective use: note generated based on sparse scratch notes.

Subjective:

- Likes [33 characters referencing appointment activities]
- [28 characters referencing behavior]
- [62 characters referencing request in appointment]
- Enjoyed [38 characters referencing appointment activities]

Objective:

- Played with [43 characters referencing appointment activities]
- [49 characters referencing sensory preferences]
- [41 characters referencing food preferences]

Analysis:

- [28 characters referencing behavior] continues to be [20 characters referencing assessment]
- [75 characters referencing request] indicates that [46 characters referencing assessment]
- Continues to [45 characters referencing sensory preferences]
- Goal of [32 characters referencing goal] remains relevant

Plan:

- Continue offering [40 characters referencing request]
- Continue playing with [43 characters referencing activities]
- Continue working on [36 characters referencing food preference recommendations]
- See [19 characters referencing next appointment]
- Continue to monitor and adjust plan as needed

Excerpt 6.2.4. Edited progress note example for effective use: note generated based on sparse scratch notes. Italicized text indicates what was added upon editing. Bolded text indicates what was moved to a different section.

Subjective:

- Likes [33 characters referencing appointment activities]
- [28 characters referencing behavior]
- [62 characters referencing request in appointment]
- Enjoyed [38 characters referencing appointment activities]
- *[25 characters referencing behavior with sauce]*
- *[24 characters referencing behavior at park]*
- *[38 characters referencing behavior with food]*
- *[32 characters referencing behavior with sibling]*
- *[37 characters referencing behavior with toys]*

Objective:

- Played with [43 characters referencing appointment activities]
- [49 characters referencing sensory preferences] but [27 characters referencing sensory preference]
- [41 characters referencing food preference]
- *enjoyed [60 characters referencing appointment activities]*
- *[70 characters referencing request]*
- *[55 characters referencing behavior in appointment]*

Analysis:

- [28 characters referencing behavior] continues to be [20 characters referencing assessment]
- [76 characters referencing request] indicates that [46 characters referencing assessment]
- Continues to [45 characters referencing sensory preference]

Plan: - Continue offering [40 characters referencing request]

- Continue playing with [43 characters referencing activities]
- Continue working on [36 characters referencing good preference recommendations]
- See [19 characters referencing next appointment]
- Continue to monitor and adjust plan as needed
- *[Place] F/U in [Month]*
- *Recommend mom to continue [75 characters referencing sensory recommendations]*
- **Goal of [32 characters referencing goal] remains relevant**

Score Type	Mean	Standard Deviation
ROUGE-1	0.706	0.172
ROUGE-2	0.584	0.253
ROUGE-L	0.623	0.239
ROUGE-LSUM	0.696	0.182

Table 6.1: Mean and standard deviation of ROUGE scores comparing custom model notes that are unedited and edited by clinicians.

6.2.2 Comparing Unedited and Edited Model Notes

The clinicians at KidsAbility were expected to read over all generated notes and edit them as appropriate. In order to compare unedited and edited model notes, and therefore gain an understanding of the degree to which clinicians were editing the notes, different ROUGE scores were calculated for each edited model note comparing it against the unedited, original note. At the time of analysis, which was after the pilot study, 64 pairs of edited and unedited notes had been collected.

Generally, ROUGE scores are used to measure similarity between a piece of text and a reference piece of text. For example, they are used as a metric for determining how closely a generated summary matches a human-written summary [13]. All ROUGE metrics provide a value between 0 and 1 with values closer to 1 meaning that there is a higher degree of similarity as measured by that particular score. ROUGE-1 scores provide a measure of the similarity based on the number of individual words that are the same, while ROUGE-2 scores provide a measure that is based on each set of 2 words, so that it captures key words and contextual pairings. ROUGE-L and ROUGE-LSUM scores provide an indicator of sentence-level similarity and similarity at the level of the entire note respectively.

The means and standard deviations of four different ROUGE metrics calculated for all model output notes compared with the final edited notes are shown in Table 6.1. All ROUGE scores were relatively high with standard deviations that suggest some variability but not to a large degree.

6.2.3 Clinician Evaluation of Generated SOAP Notes

The mean scores for the five criteria across the four different groups of SOAP notes are shown in Table 6.2, while the corresponding standard deviations are shown in Table 6.3. It can be seen that the Custom Edited notes had consistently high scores for the five criteria,

Note Type	Clear	Complete	Concise	Relevant	Organized	Average
Non-AI Note	2.237	1.944	2.569	2.611	2.055	2.28
Co-Pilot Edited	2.647	2.656	2.338	2.765	2.513	2.58
Custom Unedited	2.495	2.550	2.546	2.758	2.327	2.54
Custom Edited	2.660	2.613	2.574	2.801	2.490	2.63

Table 6.2: Mean scores assigned to SOAP notes within each note type category. The highest mean score for each criterion is bolded.

Note Type	Clear	Complete	Concise	Relevant	Organized
Non-AI Note	0.5931	0.6157	0.5680	0.5217	0.5993
Co-Pilot Edited	0.5428	0.5447	0.6542	0.3969	0.5888
Custom Unedited	0.5411	0.5173	0.5575	0.4292	0.6373
Custom Edited	0.4819	0.4606	0.5664	0.3255	0.5598

Table 6.3: Standard deviation of the scores assigned to SOAP notes averaged across the different scorers within each note type category. The lowest standard deviation for each criterion is bolded.

with the highest scores for three of the five criteria and the second highest scores for the remaining two. Meanwhile, the Non-AI notes consistently exhibit the lowest scores except for the criterion of Conciseness. Finally, looking at the standard deviations of the scores for each criterion, the custom model notes have the lowest standard deviations across all criteria, with the Custom Edited notes having the lowest for four out of five.

We performed two-factor ANOVA tests with replication on each of the five criteria, with the two factors being the clinician evaluator and the category of note. Both main factors were significant ($p \ll 0.001$) across all criteria. There were also moderate interaction effects between the two factors for all criteria.

We then performed pair-wise comparisons between scores of different note categories, using Tukey’s test to correct the significance threshold for multiple comparisons. These tests showed that there was a significant difference in mean scores between the Non-AI notes and the Custom Edited notes for all criteria except for conciseness. Conversely, there was a significant difference between the mean scores for the custom model notes and Copilot notes only for conciseness.

Note ID	Clinician ID	Time (Minutes)
2	1	9.60
5	1	6.67
6	1	7.58
7	1	5.82
8	1	9.93
11	2	8.77
12	2	9.74
15	3	14.25
16	3	6.42
17	4	<i>16.74</i>

Table 6.4: Quantitative timing data for writing progress notes with the custom model. Note ID indicates chronological order. Italicized entry indicates an outlier.

6.2.4 Timing Results

A total of 18 times were recorded including 10 in the custom model category and 8 in the manually written category. Table 6.4 shows the time spent on the process of writing notes with the custom model, while Table 6.5 shows the time spent on the process of writing SOAP notes manually. The average time spent on writing notes with the custom model was 9.55 minutes with a standard deviation of 3.5 minutes. The manually written notes took an average of 7.84 minutes with a standard deviation of 3.1 minutes. Of note is the large degree of variability in the times recorded for both categories of notes. Additionally, for one of the custom model notes, the clinician who wrote it (Clinician ID 4) commented that it was a much more complex note than usual and could be considered an outlier. Similarly, the clinician who manually wrote the 9th note (Clinician ID 2) commented that the note was shorter than usual, because it was for a virtual call that was very brief and did not contain any actual interaction with the client. These two outliers are marked as such in the tables. Excluding the time for these notes, the average time taken to write notes with the custom model is 8.76 minutes with a standard deviation of 2.6 minutes, while the average time would be 8.38 minutes with a standard deviation of 2.9 minutes.

Note ID	Clinician ID	Time (Minutes)
1	1	8.40
3	1	5.83
4	1	7.49
9	2	<i>4.02</i>
10	2	6.61
13	2	11.54
14	3	13.09
18	4	5.71

Table 6.5: Quantitative timing data for writing progress notes manually. Note ID indicates chronological order. Italicized entry indicates an outlier.

6.3 Discussion

6.3.1 Importance of Training Clinicians to use AI tool

It was found that training and clinician expectations played a large role in how useful the clinicians at KidsAbility found the model for producing SOAP notes in a time-effective manner. As mentioned in Section 6.2.1, it was clear that some of the clinicians were using the custom model more effectively and efficiently than others and when this was explored further, that these clinicians were using the model as intended while the others were not. The model was trained to produce a reasonable first draft of a SOAP note from sparse scratch notes that could then be edited. However, the majority of the clinicians in the pilot were using it such that they wrote out long and detailed scratch notes, and then were also doing a great deal of editing after the model generated a SOAP note. Thus, not only was writing out such detailed scratch notes time-consuming, but the model was not able to elaborate or contribute to the writing effectively. Instead, they were finding that it simply changed the wording, moved points around, or added text that they did not want. Meanwhile, the therapists that were using it more effectively had found that when they wrote scratch notes very quickly and without a great deal of detail, the model was able to produce a coherent first draft that they could then spend some time editing. In this way, to use the model effectively, the majority of the time spent on the process should be devoted to editing the SOAP note rather than writing out the scratch notes used as input. When coached to do this by the KidsAbility research team, all of the therapists were happier with the results and could write a high quality SOAP note in a reasonable amount of time. Therefore, it is inferred that hands-on training to use the custom model

effectively can greatly increase the benefits that it can provide.

A factor that may have contributed to the non-optimal way in which the majority of the clinicians were using the custom model is that they had already used the Co-Pilot model for writing SOAP notes previously and had biases regarding how best to use such tools. Some clinicians found that they did not like the SOAP notes that Co-Pilot provided if they did not include a great deal of detail. For instance, the clinicians specifically reported that they did not like the Analysis or Plan sections of the Co-Pilot notes if they didn't provide these themselves. Some clinicians reported only using the Subjective and Objective sections of the Co-Pilot notes and writing the Analysis and Plan sections entirely themselves.

In the future, it would likely be beneficial to provide more extensive training for clinicians on how to use the tool effectively. Coaching the clinicians through the process of using the model for themselves appears to be most appropriate for improving the efficiency with which they use the tool.

6.3.2 Editing the SOAP Notes

Based on the calculated ROUGE scores for the pairs of edited and unedited progress notes from the custom AI model, it is apparent that the clinicians are taking the time to edit the drafts of the progress notes. The relatively high scores for all ROUGE scores demonstrate that the clinicians are making changes to the notes upon editing, but the majority of the changes to the notes appear to be more minor in nature. The higher ROUGE-1 and ROUGE-LSUM scores indicate that the general similarity of the notes, on the scale of individual words and the note as a whole, is high. The somewhat lower, but still relatively high ROUGE-2 and ROUGE-L scores may indicate that most of the changes are at the level of word order and sentence structure.

The results of the clinician evaluation of progress note quality show that, while the edited custom model notes received higher scores than the unedited notes for every criteria, the unedited notes still scored consistently higher than the manually written notes except for the criterion of Conciseness. Thus, the process of editing seems to improve the quality of the note, but the quality of the unedited notes is still relatively high.

6.3.3 Generated SOAP Note Quality

The scores given to the notes generated with the custom AI model, both edited and unedited, indicate that the model is capable of producing high quality progress notes.

Compared to SOAP notes written without the use of AI, the SOAP notes written with either Co-Pilot or the custom AI model were given higher scores for all five criteria, apart from Conciseness, for which the non-AI notes received higher scores than the Co-Pilot model notes and the unedited custom model notes, but not the edited custom model notes. Therefore, it is inferred that the use of the AI models does not degrade the quality of the SOAP notes, but instead may help to produce higher quality SOAP notes.

Comparing the notes generated with the custom model to the notes generated with the Co-Pilot model, the custom model received higher scores for conciseness regardless of editing, but otherwise the scores for edited notes of both models were largely very similar. The reports of clinicians from the pilot corroborate this finding regarding how concise the notes were as they included that the Co-Pilot model generated notes that were longer and contained more “fluff,” than the custom model.

Additionally, it can be seen through the standard deviation for each criterion, averaged between different reviewers, that the custom AI model had the lowest standard deviation for all criteria. Apart from the Organization criterion, the scores for the edited and unedited custom model notes both had lower standard deviations than those of the non-AI notes and Co-Pilot generated notes. For the Organization criterion, however, the edited custom model notes still had the lowest standard deviation. It is possible that this indicates that the custom AI model notes are more consistent in quality than the other notes, but further investigation would need to be done to confirm this.

It is also important to note some of the limitations of this evaluation. Firstly, the three-point scale was somewhat limiting in that it is unable to capture finer-grained differences in quality between the notes. Thus, the differences in average scores are very small. Furthermore, the notes used in the evaluation that were generated with the custom AI model included many examples that were written by clinicians that were not using the model in the most effective manner and instead writing very long and detailed scratch notes. As a result, the findings may not be applicable to notes that are written in the more time-effective manner. For a further analysis on the data, refer to [1].

6.3.4 Comparing Timing Data

The timing data that was collected does not indicate that the use of the custom model impacts the time spent on writing progress notes. This is unsurprising given the small sample size. As discussed in Section 6.1.3, G*Power analysis indicated that a sample size of 176 for each category, with a total of 352 data points altogether, would be recommended in order to find a moderate difference in the means of the two groups. Post-hoc G*Power

Analysis shows that the power is very low for such a sample size (0.091), meaning that it is very unlikely that a statistically significant difference would have been found even if one exists. Thus, more data collection would be required to make a more definitive statement regarding whether the model tends to decrease the amount of time spent on writing progress notes.

When the average of the times is taken for each without the outliers identified by clinicians, the average time for writing notes with the model is less than 0.4 minutes longer than the average time for manually writing the notes, while the standard deviation for both categories of notes is relatively high at 2.6 and 2.9 minutes respectively. Thus, there appears to be no effect, but this does not mean that there is no difference between the two categories. Instead, it means that the sample size is currently too small to detect one.

Aside from the limitation of the small sample size, there were additional factors that could be considered limitations for this data collection process. The first of these is that there is a large degree of variation in the length and complexity of progress notes for OT treatment appointments and therefore there is a large degree of variation in how long it takes for a clinician to write a progress note regardless of whether they are using a language model or not. For example, Clinician 4 wrote one note that was an outlier in the custom model category, stating that the note took a long time to write due to the complexity of the case, the fact that there were several other clinicians and educational assistants involved, and there were many updates to go over since it had been a long time between treatment appointments (due to the summer break). On the other hand, the manual note that Clinician 4 wrote took only 5.71 minutes, which she stated was also unusual. In the case of this note, she said that she did not actually see the client, but instead exchanged emails with a parent in lieu of treatment, providing treatment advice. Additionally, there is some evidence that there is a fairly large degree of variability in how long it takes for different clinicians to write notes. For example, members of the KidsAbility innovation team report that more experienced clinicians tend to take less time writing progress notes than clinicians that are less experienced. For future experimental design, it may be beneficial to control for such factors to some extent by excluding notes that are particularly complex or particularly short in nature, or by only including clinicians with a certain number of years of experience.

A second factor of note when collecting timing data is that the clinicians expressed some degree of mental strain when writing multiple progress notes in a row. Writing five progress notes in one hour, in the case of Clinician 2, was very mentally tiring. Clinician 2 commented that she felt she was “slowing down” as the call progressed. In fact, the time taken to write each note became progressively longer for Clinician 2 regardless of whether she was using the custom model or writing the note manually. In future, such

an effect could be mitigated by offering the clinicians breaks in between writing notes, or else arranging to time the process of writing fewer notes at a time and scheduling more sessions.

Finally, another factor that could be perceived as a limitation is that despite the fact that three of the four clinicians that participated had gone through one-on-one training to use the model more effectively, they did not appear to implement the more effective strategies. While the clinicians did not share their screen for the majority of the call, they did share their scratch notes that were used to generate the draft progress note with the custom model. The scratch notes were longer and more detailed than those that the OTs had been coached previously to write. A potential reason for this could be that for most of the clinicians that participated, when they were asked to use the model for timing data collection, this was the first time they had used the model in several weeks. One participating clinician did comment, however, that she did not like providing sparser scratch notes because she thought the model was more likely to make up details to fill in gaps. She reported finding this tendency frustrating. Furthermore, two of the participating clinicians had a notable bias against using the model in general. One of the clinicians sighed in frustration when asked to use the model and paused to complain during the process of editing the progress note. In the case of this particular clinician, she was recruited to participate in the pilot study despite having a negative view on the use of AI for helping to write progress notes. She had many negative comments about Co-Pilot, but eventually found a way to use the model that worked for her. When the second phase of the pilot started, she reported being very frustrated to find that her strategy that worked with Co-Pilot no longer worked with the custom model. Together with the difficulties that she and other OTs experienced when using the web application for the custom model, this meant that her overall experience with the custom model was characterized by a lot of frustration. In the future, it would be beneficial to implement strategies that would help to mitigate such frustrations. For example, more extensive testing could be done on the web application prior to beginning the pilot study with all of the participants. Also, it would be ideal if the custom model was implemented at the same time as the Co-Pilot model as opposed to multiple weeks after the clinicians started to use Co-Pilot. Simultaneous implementation with two groups of clinicians, each using a different model, would help to mitigate the potential bias that clinicians could have in favor of the first model that they learned how to use.

Chapter 7

Conclusions

7.1 Summary

This thesis explores the development and initial evaluation of a custom large language model for pediatric occupational therapy documentation at KidsAbility. The project idea was introduced by KidsAbility with the goal of allowing clinicians to complete their documentation in less time than it currently takes to manually write this. Largely they are motivated by the idea that less time spent on documentation could reduce the incidence of clinician burnout and increase the capacity of the rehabilitation center so that they can see more clients [14]. Together, we identified the process of writing a progress note in SOAP format based on point-form scratch notes from OT treatment appointments as our target for automation and time savings. For the purpose of fine-tuning a model on this task, KidsAbility provided two datasets of historical progress notes, which were processed such that incidental PHI was removed from their text. However, these datasets did not include the corresponding scratch notes, so it was necessary to generate an additional dataset of these for model training. To ensure that these generated scratch notes would be of sufficiently high quality, we used a human-in-the-loop evaluation process to refine the generation procedure.

The Meta Llama models, in particular the smaller models of 7 to 8 billion parameters, were chosen as the base models for the custom language model based on their accessibility and proficiency at NLP tasks [38, 11]. Multiple training techniques were explored for these models including DAPT [15], which provided examples of historical progress notes as a means of adapting the model to the terminology and language patterns of pediatric OT notes, as well as fine-tuning, in which the model was shown specific examples of the

desired task, requiring scratch notes and corresponding historical progress notes. Based on comparison of the model outputs, a Llama 3 8B Instruct model fine-tuned on the progress notes and generated scratch notes was chosen for a pilot study at KidsAbility.

The pilot study involved ten clinicians using the Microsoft Co-Pilot model for writing progress notes for three weeks and then using the custom model for three weeks. Results from the study that are discussed in this thesis include qualitative findings from the pilot including subjective timing reports, as well as quantitative data in the form of ROUGE scores for comparing edited and unedited model notes, quality scores assigned by clinicians, and some timing data capturing the amount of time it takes for the clinicians to write progress notes with and without the custom model.

Overall, it was found that the generation and training procedures led to the development of a model that performs well at the required task. The model produced progress notes, both unedited and edited by clinicians, of comparable or higher quality than manually written progress notes. The impact of the model on the amount of time that it took clinicians to finish their progress notes appeared to be largely dependent on training as well as appropriate and effective use of the custom model. Therefore, the interaction between the model and the user is very influential in determining whether it accomplishes the goal of reducing the amount of time spent on documentation.

Av

7.2 Limitations

In this work, availability of the clinicians was a limiting factor in terms of involving them as domain experts in evaluation and data collection. It was not feasible, for instance, to have KidsAbility clinicians perform an evaluation on the different progress note outputs from all of the various models that were fine-tuned. Instead, it was necessary to perform this evaluation with the guidance and advice of the innovation team at KidsAbility. In the pilot study itself, the clinicians produced only 64 examples of unedited and edited progress note pairs, which were fewer than originally hoped for. Furthermore, collecting quantitative timing data to evaluate the impact of writing progress notes with the custom model was challenging given that clinicians had very limited availability for this task. Ultimately, the limited availability of clinicians meant that there was not a statistically significant finding with the quantitative timing data due to the small sample size.

Another constraint was the functionality of the web application through which the clinicians accessed the model. The clinicians faced difficulty when using the web application

due to internet connection issues, many of which may be attributed to the poor Wi-Fi connection at on-site locations. However, they also faced difficulty because of long wait times and capacity issues at the backend of the app that led to frustration. In comparison to the Microsoft Co-Pilot app that the clinicians used previously in the pilot study, these challenges are to be expected. Our web application was developed with far fewer resources and in significantly less time than that for the Co-Pilot model interface. The development of the web application is outside of the scope of this thesis, however the difficulties that the clinicians experienced when using the application impacted their impression and use of the custom model.

7.3 Future Work

There are multiple avenues of future work that could be explored, the most straightforward of which would be to collect more quantitative timing data in order to identify if the model has a statistically significant effect on the amount of time spent on writing progress notes. To collect the hundreds of data points required for this, it would perhaps be most feasible to automate the timing of the process. A timing function could be added to the web application for the custom model, and a separate, possibly related interface could be used for the clinicians to write out their notes manually with the functionality to automatically record the time spent on the process. Another avenue of future work that would be fairly straightforward would be to further fine-tune the model using examples of progress notes that were produced by the model and edited by clinicians to align better with their preferences and to address any potential errors or hallucinations. Further fine-tuning on such a dataset could teach the model how to produce progress notes that are more accurate based on the given scratch notes and more in line with clinician preferences. It is hypothesized that such additional fine-tuning on examples that include scratch notes written by clinicians instead of generated scratch notes, would benefit model performance. The fine-tuning could be performed in the same way as that of the original custom model except that it would load the already fine-tuned model and perform the supervised fine-tuning from there. The reason that this was not performed within the scope of this thesis is that time constraints and insufficient data from clinicians made the process infeasible. At least a hundred examples of edited progress notes would be needed for further fine-tuning the model in such a way. Thus, this would be an interesting direction for future work once more edited notes are available for use in training.

In order to improve upon the utility of the current custom model in the pediatric rehabilitation setting, future work could also focus on human-computer interaction. This

could involve further investigation into how clinicians prefer to use such a tool in their workflow and how the interface for using the model and the model itself can be improved to suit their needs. For instance, some clinicians expressed a wish that they could provide prompts or instructions to the model that are specific to each particular note. They also wanted to be able to provide further natural language instructions for how to improve the note after the initial generation of a progress note. These ideas are within the realm of possibility for the custom language model given the capabilities of the Meta Llama models but would require significant work on the interface aspect of the project in particular.

Additional possibilities for future work on this project that have been discussed with KidsAbility include a broadening of the scope of the model to also produce progress notes for pediatric physiotherapists and / or pediatric speech language pathologists, and to be used in other pediatric rehabilitation clinics that are affiliated with KidsAbility. An investigation into the ability of such a custom model to produce notes for multiple different disciplines within pediatric rehabilitation would be beneficial to such clinics.

Future work that KidsAbility is particularly interested in for this project is training a similar custom model to perform the task of transforming a transcript of a treatment appointment into a progress note in SOAP format. Products that perform this task currently exist, but they would prefer a private model that is trained on their data specifically and therefore would be better suited to writing progress notes in the domain of pediatric occupational therapy and that match the preferences of their clinicians. One of the reasons that this task was not explored for this thesis was that there were concerns about the appropriateness of transcripts for this task in pediatric rehabilitation given that the children that they see are typically less able than the general population to verbalize concerns and have their treatment captured in dialogue. Additionally, transcripts would be very difficult to capture in school settings such as classrooms with many students. However, KidsAbility would like to explore training their therapists to verbalize all important details during and potentially directly after an appointment so that the necessary details are present in a transcript. Another primary concern with this task at the beginning of this thesis work, was that open-source models generally did not have long enough context lengths to handle transcripts. However, recent advances have seen open-source models with much longer context lengths including the Llama 3.1 models, which support context lengths of up to 128K tokens as opposed to the earlier models' 4096 tokens [11]. Such models could more feasibly be used for taking transcripts of OT treatment appointments as input and returning draft progress notes.

References

- [1] Solomon Amenyo, Maura R. Grossman, and Daniel G. Brown. Keut: Assessment of ai-generated pediatric rehabilitation soap note quality, 2024. [Forthcoming Fall 2024].
- [2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. (arXiv:2005.14165), July 2020. arXiv:2005.14165 [cs].
- [3] Helen Buchanan, Jennifer Jelsma, and Nandi Siegfried. Practice-based evidence: Evaluating the quality of occupational therapy patient records as evidence for practice. *South African Journal of Occupational Therapy*, 46(1):65–73, April 2016.
- [4] Susan M. Cahill and Stephanie Beisbier. Occupational therapy practice guidelines for children and youth ages 5-21 years. *AJOT: American Journal of Occupational Therapy*, 74(4):7404397010p1–7404397010p1, July 2020.
- [5] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3):39:1–39:45, March 2024.
- [6] Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models via reading comprehension. (arXiv:2309.09530), February 2024. arXiv:2309.09530 [cs].

- [7] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [8] Gloria Frolek Clark and Karrie L. Kingsley. Occupational therapy practice guidelines for early childhood: Birth-5 years. *AJOT: American Journal of Occupational Therapy*, 74(3):7403397010p1–7403397010p1, May 2020.
- [9] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. (arXiv:2305.14314), May 2023. arXiv:2305.14314 [cs].
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. (arXiv:1810.04805), May 2019. arXiv:1810.04805 [cs].
- [11] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Young, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira,

Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Conguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenxin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz,

Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damla, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabza, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vitor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihalescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi,

- Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models. (arXiv:2407.21783), August 2024. arXiv:2407.21783 [cs].
- [12] Salvatore Claudio Fanni, Maria Febi, Gayane Aghakhanyan, and Emanuele Neri. *Natural Language Processing*, page 87–99. Springer International Publishing, Cham, 2023.
 - [13] Nikolaos Giarelis, Charalampos Mastrokostas, and Nikos Karacapilidis. Abstractive vs. extractive summarization: An experimental review. *Applied Sciences*, 13(1313):7620, January 2023.
 - [14] Christopher A. Goffredo, Patricia Bowyer, Helena I.S. Reis, and John Humphrey. Pediatric occupational therapists and occupational stress: A scoping review. *Occupational Therapy In Health Care*, 0(0):1–15, 2022.
 - [15] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don’t stop pre-training: Adapt language models to domains and tasks. (arXiv:2004.10964), May 2020. arXiv:2004.10964 [cs].
 - [16] Md. Asraful Haque and Shuai Li. Exploring chatgpt and its impact on society. *AI and Ethics*, February 2024.
 - [17] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. A simple language model for task-oriented dialogue, May 2020.
 - [18] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. (arXiv:2106.09685), October 2021. arXiv:2106.09685 [cs].
 - [19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. (arXiv:2001.08361), January 2020. arXiv:2001.08361 [cs, stat].
 - [20] Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. Synthetic data generation with large language models for text classification: Potential and limitations. (arXiv:2310.07849), October 2023. arXiv:2310.07849 [cs].
 - [21] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477, December 2022.

- [22] Yu Meng, Martin Michalski, Jiaxin Huang, Yu Zhang, Tarek Abdelzaher, and Jiawei Han. Tuning language models as training data generators for augmentation-enhanced few-shot learning. In *Proceedings of the 40th International Conference on Machine Learning*, page 24457–24477. PMLR, July 2023.
- [23] Meta. Meta llama 3: Model cards and prompt formats.
- [24] Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp, December 2021.
- [25] Josh Nguyen and Christopher A. Pepping. The application of chatgpt in healthcare progress notes: A commentary from a clinical and research perspective. *Clinical and Translational Medicine*, 13(7):e1324, July 2023.
- [26] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madeleine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew

Kondrich, Aris Konstantinidis, Kyle Kopic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report. (arXiv:2303.08774), March 2024. arXiv:2303.08774 [cs].

- [27] Sajjan B. Patel and Kyle Lam. Chatgpt: the future of discharge summaries? *The Lancet Digital Health*, 5(3):e107–e108, March 2023.
- [28] A. Perez-Castro, M. R. Martínez-Torres, and S. L. Toral. Efficiency of automatic text generators for online review content generation. *Technological Forecasting and Social Change*, 189:122380, April 2023.

- [29] Vivek Podder, Valerie Lew, and Sassan Ghassemzadeh. *SOAP Notes*. StatPearls Publishing, Treasure Island (FL), 2024.
- [30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.
- [31] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI*, 2018.
- [32] Amazon Web Services. What is amazon comprehend? - amazon comprehend.
- [33] Amazon Web Services. What is amazon s3? - amazon simple storage service.
- [34] Amazon Web Services. What is amazon sagemaker? - amazon sagemaker.
- [35] Amazon Web Services. Sagemaker studio administration best practices, April 2023.
- [36] Fatemeh Shah-Mohammadi and Joseph Finkelstein. Extraction of substance use information from clinical notes: Generative pretrained transformer-based investigation. *JMIR Medical Informatics*, 12(1):e56243, August 2024. Company: JMIR Medical Informatics Distributor: JMIR Medical Informatics Institution: JMIR Medical Informatics Label: JMIR Medical Informatics publisher: JMIR Publications Inc., Toronto, Canada.
- [37] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. (arXiv:1409.3215), December 2014. arXiv:1409.3215 [cs].
- [38] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan

- Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, July 2023.
- [39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
 - [40] Sheng Wang, Zihao Zhao, Xi Ouyang, Qian Wang, and Dinggang Shen. Chatcad: Interactive computer-aided diagnosis on medical image using large language models. (arXiv:2302.07257), February 2023. arXiv:2302.07257 [cs, eess].
 - [41] Michael Wornow, Yizhe Xu, Rahul Thapa, Birju Patel, Ethan Steinberg, Scott Fleming, Michael A. Pfeffer, Jason Fries, and Nigam H. Shah. The shaky foundations of large language models and foundation models for electronic health records. *npj Digital Medicine*, 6(1):1–10, July 2023.
 - [42] Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. Bag of tricks for training data extraction from language models, February 2023.
 - [43] Hongjian Zhou, Fenglin Liu, Boyang Gu, Xinyu Zou, Jinfa Huang, Jinge Wu, Yiru Li, Sam S. Chen, Peilin Zhou, Junling Liu, Yining Hua, Chengfeng Mao, Chenyu You, Xian Wu, Yefeng Zheng, Lei Clifton, Zheng Li, Jiebo Luo, and David A. Clifton. A survey of large language models in medicine: Progress, application, and challenge. (arXiv:2311.05112), May 2024. arXiv:2311.05112 [cs].

APPENDICES

Appendix A

Additional Examples of SOAP Notes Generated by the Different Models

A.1 Full Set of Generated Progress Notes

Excerpt A.1.1. Historical scratch note example.

Met with Eas [17 characters referencing staff names]

[36 characters referencing toilet paper dispenser]

Staff [94 characters referencing impulsivity and communication issues].

Discussed [22 characters equipment placement], [171 characters referencing positioning advice]

Ct [38 characters referencing participation] – [7 characters referencing staff behavior]

Reviewed [78 characters referencing specific equipment positioning]

Ct [115 characters referencing participation and abilities]

Staff [27 characters referencing staff behavior] and [57 characters referencing response to OT recommendations]

Excerpt A.1.2. Historical progress note example.

OT school visit

Late entry for [DATE]

OT met with EA's [18 characters referencing staff names] to review [16 characters referencing care task].

O- [37 characters referencing toilet paper dispenser] as per OT recommendations. Staff [120 characters referencing impulsivity and communication issues] when providing training. OT discussed rational for training and ct needs. Staff [16 characters referencing impulsivity] and [48 characters referencing client participation]. OT recommended [122 characters referencing recommendations].

OT stressed the following [19 characters referencing care task] -

- [186 characters referencing positioning advice]
- [126 characters referencing OT recommendation]
- [39 characters referencing communication advice] – [28 characters referencing OT recommendation]
- [122 characters referencing positioning advice].

A- Staff [39 characters referencing impulsivity and communication issues] and [39 characters referencing response to OT recommendations]. Ct [45 characters referencing participation]. Following OT recommendations as written on [Task] documents provided to school is essential for safety of [NAME] and caregiver during [Task]. OT reviewed observations with CRT following visit – [77 characters referencing OT recommendations].

PLAN - school to contact OT with any additional concerns. OT to follow up as needed.

[NAME], OT Reg. (Ont)

Excerpt A.1.3. SOAP Note generated by pre-trained Llama 2 model with the GoldCare training dataset (no fine-tuning). Italicized text indicates tokens that should not be present in the response. Bolded text indicates that the text is not correct based on the scratch note.

<<SYS>>

Here is a Progress Note in SOAP format.

This is a Progress Note - OT from [DATE]:

S: School visit today (Visit [DATE]). OT met with school staff (EAs: [17 characters referencing staff names]) and student.

O: EA shared that she was not aware of [NAME]'s [Task] protocol (see Visit Note in Documentation for recommendations provided). She followed up with staff from [Clinic] to discuss.

A: OT and EA discussed that [NAME] needs more time to settle into the classroom. We need to give her time to adjust to the environment, staff and peers. We need to build trust and consistency between staff and [NAME]. See Visit Note in Documentation for recommendations provided. Two copies of the visit note were left at school. Staff was instructed to send a copy home to family/guardian and keep a copy for their records. All appropriate COVID-19 precautions taken as per Board's and KidsAbility's requirements.

P: Next visit booked with school. OT to continue to provide support.

[/INST]

Excerpt A.1.4. SOAP Note generated by pre-trained and LoRA fine-tuned Llama 2 model with the GoldCare training dataset. Bolded text indicates that the text is not correct based on the scratch note. The note is missing many details and is potentially misleading.

This is a Progress Note - OT from [DATE]:

[44 characters referencing toilet paper dispenser] due to [8 characters referencing impulsivity] by staff. [56 characters referencing positioning assessment (Somewhat Incorrect)].
Toilet paper was dispensed from correct placement.

P: No follow up required. Continue with schedule.

Excerpt A.1.5. SOAP Note generated by pre-trained and fully fine-tuned Llama 2 model with the GoldCare training dataset. Bolded text indicates that the text is not correct based on the scratch note.

This is a Progress Note - OT from [DATE]:

Late entry for [DATE]

OT visit to the home to [34 characters referencing toilet paper dispenser (Incorrect)]. **This was moved by staff from the previous day and OT was asked to go to the home to review the positioning for [13 characters referencing equipment].**

S: Staff reported that they have **moved the [99 characters referencing equipment positioning (Incorrect)].**

O: Reviewed positioning with staff and advised **they think that [75 characters referencing position assessment (Incorrect)]. Discussion with staff and advised they will need to [114 characters referencing OT recommendation (Incorrect)]. This is [128 characters referencing positioning assessment (Incorrect)].**

A: Staff is [32 characters referencing impulsivity] and [61 characters referencing response to OT recommendations].

P: OT to continue to provide support and progress strategies **for Eas to access [29 characters referencing care tasks (Incorrect)] at school.**

Excerpt A.1.6. SOAP Note generated by LoRA fine-tuned Llama 2 model with the GoldCare training dataset. Several lines were edited out of the note to fit it onto a single page. There were dozens of lines of [NAME] OT [DATE] repeated at the end of the note that were edited out as well.

SOAP Notes: [DATE]
 OT: [NAME]
 P: [NAME], [NAME]
 S: [NAME] OT [DATE]
 O: [NAME] & [NAME] met with Eas [17 characters referencing staff names] OT [DATE]
 * [NAME] is now accessing [26 characters referencing toilet paper dispenser (Incorrect)].
 * [NAME] is using [38 characters referencing equipment (Incorrect)]. [NAME] is in program to [33 characters referencing program (Incorrect)]. [NAME] is using [14 characters referencing equipment (Incorrect)].
 * [NAME] staff report [70 characters referencing impulsivity and communication issues (Somewhat Incorrect)].
 * OT discussed with staff about positioning [NAME] for [Task].
 * discussed placement of [9 characters referencing equipment].
 * reviewed need to [88 characters referencing equipment positioning (Somewhat Incorrect)].
 * discussed [47 characters referencing equipment positioning].
 * **discussed placement of [82 characters referencing equipment positioning (Incorrect)].**
 * discussed that [68 characters referencing equipment positioning].
 * reviewed positioning with staff, advised them to be thoughtful of this position and to make sure [NAME] is safe in her position.
 * OT discussed that Ct [38 characters referencing participation] – [7 characters referencing staff behavior].
 * OT reviewed need to [68 characters referencing positioning advice].
 * CT was [26 characters referencing participation].
 * OT observed that CT [52 characters referencing participation and abilities].
 * CT [27 characters referencing participation].
 * OT discussed with staff the importance of following OT recommendations and reminders.

S:
 * [NAME] is a school-based OT in [NAME] Region.
 * [NAME] sees [NAME] for OT.
 * [NAME] is [AGE (Incorrect)], in [Grade (Incorrect)].

[1231 characters referencing repeated points]

O: [NAME] OT [DATE]
 O: [NAME] OT [DATE]
 [NAME] OT [DATE]
 [NAME] OT [DATE]
 [NAME] OT [DATE]
 [NAME] OT [DATE]
 [NAME] OT [DATE]
 [NAME] OT [DATE]
 [NAME] OT [DATE]
 [NAME] OT [DATE]

Excerpt A.1.7. SOAP Note generated by fully fine-tuned Llama 2 model with the GoldCare training dataset. Bolded text indicates that the text is not correct based on the scratch note.

This is a Progress Note - OT from [DATE]:

OT coaching session at KidsAbility. Eas [NAME] attended with [28 characters referencing staff names (Incorrect)].

S/O: Updates: [36 characters referencing toilet paper dispenser]. Staff is [95 characters referencing impulsivity and communication issues].

Discussed [22 characters equipment placement], [171 characters referencing positioning advice]

Ct [37 characters referencing participation] – [7 characters referencing staff behavior].

OT reviewed [78 characters referencing specific equipment positioning] Ct [115 characters referencing participation and abilities]

Staff [27 characters referencing staff behavior] and [57 characters referencing response to OT recommendations]

Actions: OT provided education and positioning information to staff and Ct.

Plan: OT to continue to provide support as needed. **OT to review positioning and recommendations with [NAME] next week. OT to send suggestions to [NAME] for how staff can best position Ct to support transitions and participation in play.**

[NAME] OT Reg. (Ont.)

Excerpt A.1.8. SOAP Note generated by fine-tuned Llama 2 model with the AlayaCare training dataset. Bolded text indicates that the text is not correct based on the scratch note.

Subjective

-The [34 characters referencing toilet paper dispenser (Incorrect)]

Objective

-OT met with Eas [17 characters referencing staff names]

-Staff were [94 characters referencing impulsivity and communication issues]

-Discussed [22 characters referencing equipment placement], [172 characters referencing positioning advice]

-Ct [38 characters referencing participation] – [7 characters referencing staff behavior]

-Reviewed [78 characters referencing specific equipment positioning]

-Ct [115 characters referencing participation and abilities]

-Staff [27 characters referencing staff behavior] and [56 characters referencing response to OT recommendations]

Analysis

- [NAME] is able to **communicate their needs better** and [47 characters referencing participation]

- Staff are not implementing recommendations as outlined and appear to be **overstaffing** and rushing when supporting [NAME]

Plan

-Follow up visit to review progress

-OT to send email to staff summarizing recommendations discussed today

-Continue to work towards [NAME]'s independence in [Task]

-[NAME] will benefit from increased focus on positioning and staff education on [Task] equipment and [Task]

Excerpt A.1.9. SOAP Note generated by fine-tuned Llama 3 model with the AlayaCare training dataset. Bolded text indicates that the text is not correct based on the scratch note.

Subjective:

[28 characters referencing toilet paper dispenser]
Staff [80 characters referencing impulsivity and communication issues]
[22 characters referencing staff behavior] through positioning and [Task]
Ct [56 characters referencing preferences (Somewhat Incorrect)]
[85 characters referencing participation and abilities]
Staff show little interest in OT recommendations and reminders

Objective:

Reviewed [38 characters referencing equipment positioning]
Discussed [107 characters referencing specific equipment positioning]
Staff to review and follow up on positioning and [Task]
Ct [52 characters referencing participation and abilities]

Analysis:

Eas [NAME] and [NAME] require more attention to detail and understanding of positioning and [Task] strategies
Impulsive behavior needs to be addressed through redirection and positive reinforcement
Ct [39 characters referencing participation], **but may require ongoing support and reminders**
Staff need to show more interest in OT recommendations and reminders

Plan:

Continue to monitor positioning and [Task] strategies
Review positioning and [Task] with staff and provide feedback
Discussed need for [98 characters referencing specific equipment positioning]
Ct to [30 characters referencing participation] with support