Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

UKE HAMBURG

Fine-Tuning Small Large Language Models for Patient Trial Matching in Precision Medicine

—

Master's Thesis

Kevin Kraus
7512566

at
House of Computing and Data Science (HCDS)
Language Technology Group (LT) in cooperation with the Institute for Applied Medical Informatics (IAM)

Reviewer:	Prof. Dr. Chris Biemann
Supervisor (UHH):	Saba Anwar
Supervisor (UKE):	Christopher Gundler

28.02.2024

Abstract

This work investigates the task of automated patient-trial matching through fine-tuning *Llama2* chat (13B) proposing *TrialLlama*.
TrialLlama is trained on Clinical Trials from a snapshot of the *clinicaltrials.gov* website and synthetic patient descriptions provided by the TREC Clinical Trials track, using a supervised classification approach.

Two primary tasks are explored with this fine-tuned model: 1) patient-trial classification, where the model categorises patient-trial pairs into one of three labels (*eligible, excluded, irrelevant*), and 2) reasoning, where it extracts and discusses the eligibility criteria from a clinical trial to determine a patient's eligibility to get enrolled in the corresponding trial. In the patient-trial matching task treated as a binary classification, combining the two negative labels into one class, *TrialLlama* achieved an accuracy of 0.813 and an F1 score of 0.883. For the original three-label classification task, *TrialLlama* achieved accuracy and F1-scores of 0.634 and 0.530, respectively. Notably, *TrialLlama* excelled in the reasoning task, exceeding *Llama2* by 0.640 in precision and 0.666 in accuracy. Despite being fine-tuned for classification, *TrialLlama* demonstrated proficiency in extracting eligibility criteria and assessing a patient's eligibility concisely and logically.

However, several limitations are identified, including fine-tuning difficulties due to dataset limitations, a bias towards extracting inclusion criteria, hallucination issues, and comparability to other systems. Nevertheless, *TrialLlama* and its open-source codebase hold promise for advancing research in automated patient-trial matching and AI-driven medical assistants.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
AUROC	Area Under the Receiving Operator Curve
BERT	Bidirectional Encoder Represent from Transformers
CT	Clinical Trial
DNN	Deep Neural Network
EC	Eligibility Criteria
EHR	Electronic Health Records
ETL	Extract Transform Load
IAM	Institute for Applied Medical Informatics
IR	Information Retrieval
LLM	Large Language Model
LoRA	Low-Rank Adaptation
LT	Language Technology
LtR	Learning to Rank
ML	Machine Learning
MTB	Molecular Tumor Board
NCT	National Clinical Trial
NER	Named Entity Recognition
NLP	Natural Language Processing
NN	Neural Network
PEFT	Parameter-Efficient Fine-Tuning
RAG	Retrieval-Augmented Generation
RLHF	Reinforcement Learning by Human Feedback
SOTA	State-of-the-Art
TREC	Text REtrieval Conference
UKE	Universitätsklinikum Eppendorf

1. Introduction

This chapter introduces the motivation in Section 1.1 and the problem statement in Section 1.2. From there, this work's approach is defined in Section 1.3 and two research questions derived in Section 1.4, to guide the succeeding chapters of this work. Additionally, information about this work's collaboration partner is given in Section 1.5. Lastly, a short overview of this work's overall structure is given in Section 1.6.

1.1. Motivation

Patient-trial matching is the task of finding qualified patients for a Clinical Trial (CT). The importance of enrolling a patient in a CT comes from the opportunity to provide them with the best possible care, relieve them or their insurance of the financial burden, and simultaneously contribute to medical research.
The matching task is based on a patient's Electronic Health Record (EHR) and its comparison to information stated in a CT. Therefore, every CT record contains a large number of fields containing relevant information about the planned trial. Some fields are structured, some semi-structured and some are unstructured. These data fields are specified by the *Protocol Registration Data Element Definition*[1] provided by the *clinicaltrials.gov* organisation, containing required and optional fields. Besides meta information like study title, type, status, duration, and demographic information, the listed eligibility criteria (EC) play a central role in the patient-trial matching process. The ECs are provided in an unstructured, free-text format and defined as a bulleted list of relevant inclusion and exclusion factors by protocol. To determine a patient's eligibility, some aspects of the meta information, the demographic information all of the ECs have to be considered, of which all inclusion criteria have to be met and none of the exclusion criteria.

Patient-trial matching plays an important role in oncology since tumours consist of highly individual and unique mutations, varying from patient to patient, which makes treatment approaches highly dependent on the patient's genomic profile, typically requiring CTs to test and develop new approaches suitable for this certain group of patients. Additionally, cancer patients being enrolled in a CT are linked to a higher overall survival rate [6]. This genomic profiling is part of the precision (or personalised) medicine paradigm, a data-driven and patient-centred approach, that aims to combine up-to-date patient data, including clinical, lifestyle, genetic and further biomarker information to derive possible actions for highly personalised therapy, maximising the treatment efficacy while minimising adverse effects [7].

A hands-on example for patient-trial matching in oncology can be seen in the Molecular Tumor Board (MTB) approach. MTB is a treatment model in the domain of precision medicine bringing together a plenum of experts from different domains to discuss possible next steps for a patient who did not respond to standard-of-care therapy based on genetic analysis [8]. Taking a look at the specific MTB process for this work's collaboration

[1] https://clinicaltrials.gov/policy/protocol-definitions, accessed 10.02.2024 15:04

partner the Institute for Applied Medical Informatics (IAM) at the Universitätsklinikum Eppendorf (UKE) in Hamburg, Germany, the MTB process consists of multiple steps such as the transmission of the patient's EHR, running molecular diagnostics, the creation of a report for the plenum's meeting, as well as the crafting of a post-discussion document including the patient's next treatment actions. Typically, a single oncologist responsible for the patient, deep-dives into the patient's EHR and genomic mutations profile to find evidence in the form of the relevant publication and CTs to craft the required report, which is used as a plinth for the upcoming discussion (Figure 1.1 elaborated by Lauk, Peters, Velthaus, *et al.* [1]).

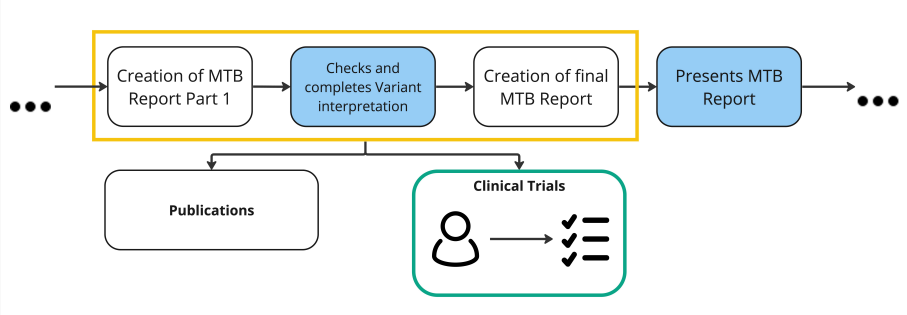

Figure 1.1.: The most important MTB process steps for this work adapted from Lauk, Peters, Velthaus, *et al.* [1]

Within this evidence search process, which is typically conducted manually by the doctor, patient-trial matching plays a major role. In the first step, a pre-filtering of CTs has to be carried out to retrieve a set of CTs matching the patient's demographics and general disease description. For example, a CT on breast cancer is irrelevant for a patient who suffers from pancreatic cancer. After coarse pre-filtering, in the second step, the oncologist goes through the relevant CTs, extracting all inclusion and exclusion criteria, which are then compared one by one to the patient's EHR. All of the inclusion criteria and none of the exclusion criteria have to be met. If any of these two conditions is not complied with, the patient is excluded from the trial. It is important to emphasise that finished CTs are of the same relevancy as recruiting ones, since they may contain relevant results for the patient at hand.

This search for suitable trials is a tough task because of multiple reasons:
1) Even though some structure is provided, the highly important EC field is unstructured free-text,
2) the number of individual inclusion and exclusion criteria within a CT varies in number from CT to CT,
3) they can range from only a few, simply written criteria to long lists of 20 or more criteria, written in different levels of complexity depending on the individual who was in charge writing them.

Hence, extracting the ECs, comparing their semantic content to the patient's EHR and considering the demographics as well as other meta-information from the trial can quickly introduce errors, leading to missed chances for a patient's therapy in conclusion. Because

of all these reasons, this task requires substantial time and concentration efforts by the physician.

Additionally, even finding possibly relevant trials in the first pre-filtering step is not a trivial task since a huge database with hundreds of thousands of trials is available, of which many are structured in old protocol standards, plus being paired with keywords from the ever-changing medical terminology, leading to a vast amount of possibly relevant keywords. This makes the search process even more challenging since it requires the doctors to craft and test different sets of queries and filters, as well as adjust to alternating CT structures to find their way through the abundance of online available CTs. This increases the chances of missing something, highlighting the need for some technical assistance to save time, decrease error and increase the patient's odds for trial enrolment.

1.2. Problem Statement

This work aims to investigate the utilisation of open-source small-sized LLMs for the specific task of patient-trial matching developing a prototype utilisible in practice by physicians. Language Technology (LT) and especially Large Language Models (LLMs) can aid the physicians in the patient-trial matching process, facilitating the manually conducted patient-trial matching, saving time, money and improve the overall enrolment results. This can be achieved by automatically matching patient-trial pairs, with a focus on relevant patient information and the ECs in the CT. Previous work was only conducted with big, mostly non-open source LLMs to tackle the problem of automated patient-trial matching. These solutions are not practical for real clinical usage due to 1) not considering limited hardware capacities for fine-tuning and running the models in a clinical environment, 2) not openly available model weights for local fine-tuning, 3) no offline use possible, and most importantly, 4) for privacy issues.
Besides the automated classification of suitable CTs, the chat capabilities of the LLM should be preserved, to allow for an assistant system with human-AI interaction. This is of high importance since the doctor in charge has to approve the final decision, rendering a sole classification impractical.

Hence, the prototype is expected to perform strongly in understanding, extracting and structuring the inclusion and exclusion criteria, since only the inclusion criteria have to be matched while the exclusion criteria should not be matched by a patient's EHR. Moreover, Other than the information provided in EC, the extraction task should also focus on other relevant parts such as the title and short description, etc to respond in a meaningful way, that is useful and comprehensible by the physician. Further, such a prototype should have good generalisation capabilities perspectively, to allow for reliable results in everyday clinical practice patient-trial matching.

1.3. Approach

The core idea is to fine-tune an open-source small-sized LLM to compare a patient's EHR to a set of ECs, providing a physician with either a simple classification of the patient's

eligibility or a thorough extraction and reasoning of each criterion. Given the available dataset provided by the Clinical Trial of the Text REtrieval Conference (TREC)[9], on the patient's side, only a short patient description is used, which is typically part of an EHR resembling the doctor's letter. This patient description gives a short, unstructured summary of the patient at hand. Two tasks are defined based on the problem statement, utilising the same prototype fine-tuned on the available datasets:
1) Patient-Trial Classification Task: Given a patient's description and a CT, the classification task can be defined as mapping the patient-trial pair (p, t) from dataset D to one of the three possible classes, denoted as

$$f : P \times T \rightarrow \{eligible, excluded, irrelevant\}$$

2) The Patient-Trial Reasoning Task: Given a patient's description and a CT, the ECs are extracted by the model and classified step-by-step with a final verdict, allowing for a better understanding of the model's classification output. This can be defined as analysing the patient's description in extracting clinically relevant aspects $R = (r_1, r_2, ...r_i)$ to match them against a set of extracted criteria $C = (C_{inclusion}, C_{exclusion})$, consisting of inclusion and exclusion factors of which all the $C_{inclusion}$, and none of the $C_{exclusion}$ have to be met, respectively.

The prototype is based on the open-source *Llama2 chat* model by Meta. Given the available datasets, the prototype is fine-tuned in a supervised approach, providing the model with a prompt, consisting of an instruction, the patient-trial pair and the output label. The focus of this work lies specifically in the extraction and comparison of ECs relevant to the patient's description. Hence, other meta-information such as *Gender* and *Location* was omitted, simply to reduce complexity, focusing on the most complex part of patient-trial matching: The manual extraction and comparison of relevant patient information and ECs.

First of all comprehensive literature analysis is conducted to get a grasp on current State-of-the-Art (SOTA) systems and approaches. The publicly available datasets utilised in these SOTA systems are then harnessed to develop this work's prototype to answer the question of eligibility for a given patient-trial pair. Within the experimentation, hyperparameter tests are conducted and the best results are used for the final model fine-tuning. The final prototype is then evaluated by addressing the classification task (quantitatively) and the reasoning task (qualitatively), comparing them to similar systems and the base model (*Llama2)*. This work's codebase is open-source and freely usable, to engage other projects related to this and similar work[2].

Small-sized LLMs

For the scope of this work let's define a small-sized LLM as an LLM with a parameter count $\leq$ 13 billion.

[2]https://github.com/UKEIAM/TrialLlama/tree/main/

1.4. Research Questions

Based on this work's approach to leverage small-sized LLMs fine-tuned for patient-trial matching, the following two research questions can be derived:

> **RQ 1**: Can a small-sized open-source LLM be fine-tuned to achieve feasible results in classifying patient's eligibility among other patient-trial matching systems?
>
> To evaluate this RQ, the fine-tuned model is benchmarked with 1) common performance metrics and 2) TREC Clinical Trials track teams from 2021 and 2022 via the official TREC evaluation script[3].
>
> **RQ 2**: Can a small-sized fine-tuned LLM give reasonable rationals for its patient eligibility classification for patient-trial matching?
>
> The evaluation of RQ 2 is conducted qualitatively. The pre-trained and fine-tuned models are used for inference to generate the required responses. Then, in the first step, two raters at the IAM rated the model's responses guided by an annotation guideline developed for this work. In the second step, the pre-trained model's and fine-tuned model's responses are further analysed manually by counting the number of overall ECs, extracted ECs, correctly classified ECs and correctly classified patient-trial pairs, allowing for response and more fine-grained performance insights.

1.5. Collaboration

Some of the insights this work was built on, were derived from the cooperation with the UKE, working closely with the oncologists of their MTB team. Through the analysis of observations and question-answering sessions with the doctors in charge of the MTB preparation and other experts in oncology at UKE important aspects of the oncologist's report creation process were derived. The most important ones are that 1) the search process for evidence is a weary, manual task, costing a lot of time and effort, and being prone to error, and b) any kind of pre-filtering of results would be beneficial for the doctors. In this work's case, the patient-trial matching task.

Further, the proposed prototypes are planned to be refined, adjusted and integrated within the *Knowledge Connector*, a platform developed by the German Cancer Research Center and the National Center for Tumor Diseases to support physicians in making therapy recommendations for patients based on molecular biomarkers and relevant clinical data. The Knowledge Connector is mainly used for MTB preparation. The primary objective is the search for suitable therapy approaches[4].

[3] https://github.com/usnistgov/trec_eval, accessed 01.12.2023 17:38
[4] https://www.dkfz.de/de/clinical-trial-office/knowledgeconnector.html, accessed 01.12.2023 18:11

1.6. Thesis Structure

This work is structured in eight chapters. Besides the motivation and problem statement in Chapter 1, an introduction to all required prerequisites is given in Chapter 2, continuing with the related work in Chapter 3, providing the reader with a summary of relevant literature related to this work. Chapter 4 describes the experimental design, discussing this work's datasets, the prototype fine-tuning, some implementation details and the discussion of the results. In Chapter 5, the technical adjustments and the results for the reasoning task are summarised and discussed. Chapter 6 describes future work for this work's objectives. Lastly, this work is concluded in Chapter 7 by revisiting the research questions, stating theoretical and practical implications and closing final words.

2. Theoretical Foundations

This chapter aims to create a foundation with background information for the reader, in introducing all relevant topics related to this work's main research objectives. In Section 2.1 a short introduction to Deep Neural Networks (DNNs) is given. Section 2.2 continues with an overview of the notion of LLMs, giving details about base model training, fine-tuning, prompting strategies and general LLM drawbacks. Section 2.3 dives into the concept of relevance ranking in information retrieval (IR), introducing classical and neural network (NN) based approaches.

2.1. Deep Learning

2.1.1. Overview

With advances in ML and growing computational possibilities due to smaller, more capable hardware deep learning became very popular thanks to its superb performance in different domains such as computer vision, speech recognition, Natural Language Processing (NLP), machine translation and many more. Generally, deep learning is defined as a subset of ML based on artificial neural networks, where *deep* refers to the architectural choice of using two or more NN layers, allowing for the learning of higher-level features and capturing more complex knowledge representations.

DNNs are most prominently associated with deep learning. They are simply NNs containing two or more hidden layers, unlocking the network's high learning capabilities utilised in many different domains. These DNNs were made possible with the exploration of the *Backpropagation* technique, solving some emerging problems of NNs explored by Rumelhart and Mcclelland [10], an important foundation for updating the node weights of NNs after each training cycle was established, introducing a new era of more complex and deep NNs consisting of millions of neurons.

Figure 2.1 depicts all relevant components of a simple NN. In area (A) an *artificial neuron*, defined as the smallest computational unit of an NN, with multiple inputs $\{x_1, x_2, x_3\}$ and a single output, utilising the Sigmoid activation function depicted in the top-left corner (C) can be seen. A more complex architecture with multiple, one hidden layer and one output is displayed in (B), utilising different activation functions for the different layers. From left to right the Sigmoid, Tanh, Identity and ReLU functions are shown in (C).

Besides DNNs, other common architectures related to deep learning and NNs are Convolutional Neural Networks, Recurrent Neural Networks and Transformers.

2.1.2. Supervised and Unsupervised Learning

Two types of ML algorithms can be abstracted: Supervised and Unsupervised learning. The major difference between both algorithms is the presence or absence of labels (the ground truth) in the data used for training the model [11].

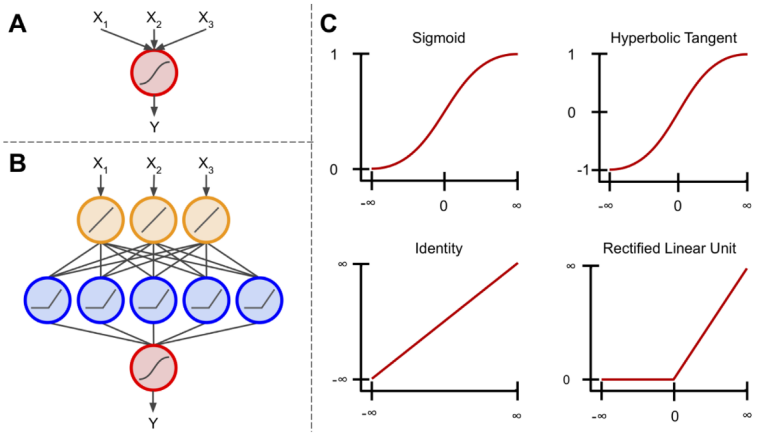

Figure 2.1.: Primary components of a Neural Network described by Choi, Coyner, Kalpathy-Cramer, *et al.* [2].

Supervised learning is generally used for discriminative tasks, mapping an input to its respecting output. This allows us to solve a broad set of classification problems, where input and output are well-defined. Typically, Backpropagation is used as the learning method. The biggest issue of supervised learning is stated by the *bias-variance tradeoff* [12]. High variance with low bias occurs when the trained model learns the data by heart (overfitting) without really understanding the underlying concepts, leading to high data sensitivity and no knowledge transfer capabilities. On the contrary, a low variance with a high bias occurs when the model was trained too shallowly (underfitting), only having a general understanding of the underlying problem.
Important to mention is the notion of *Self-Supervised* learning. Here, inherited data structures or relationships are leveraged to derive meaningful training signals, being somewhat similar to an externally provided label in the supervised learning setting, yet performing inferior compared to their fully supervised counterparts [13].
Nevertheless, self-supervised learning comes in handy for problems where massive amounts of data are ingested for model training, such as it is in the case of LLMs.

Unsupervised learning on the other hand is usually used for generative tasks when no label is available beforehand. Examples for unsupervised learning are anomaly detection, data clustering or image/ text generation. Contrary to supervised learning, where Backpropagation is predominantly used, unsupervised learning relies on other learning techniques such as *Contrastive Learning*, *Maximum Likelihood* and *Variational Inference*, just to name a few. Most of these methods rely on statistical properties, trying to learn and mimic the distribution of the provided data to allow for the generation of new data, sampled from the learned distribution [14].

2.1.3. Transformer Architecture

The transformer architecture introduced by Vaswani, Shazeer, Parmar, *et al.* [3] in their paper "Attention is All You Need" states a pivotal event for language technology in 2017. Figure 2.2 shows the Transformer architecture proposed in the original paper. Its archi-

tecture is inspired by the simple idea of the encoder-decoder (Seq2Seq) architecture, most prominently used in machine translation mapping an input to an output.

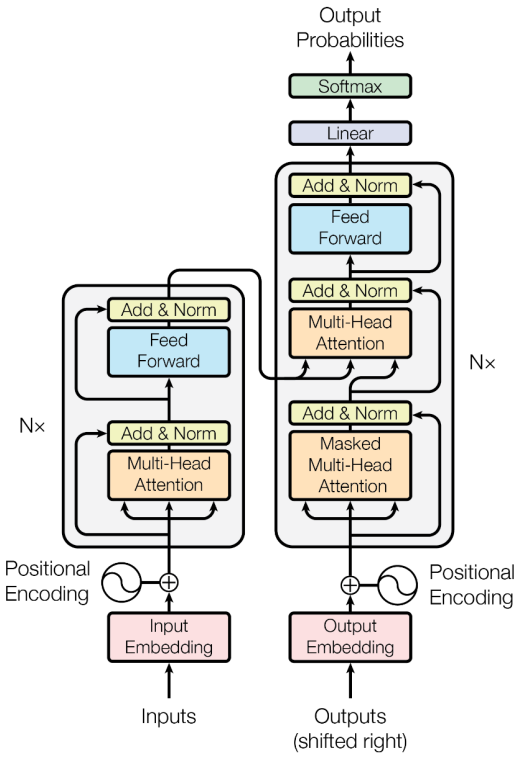

Figure 2.2.: Transformer architecture by Vaswani, Shazeer, Parmar, *et al.* [3].

The Transformer architecture consists of an encoder and a decoder block, which contain different sub-components based on concepts like positional encoding, attention mechanisms, normalisation- and feed-forward layers. Lastly, before outputting anything, a linear layer followed by a softmax layer is utilised, simply transforming the results to a human-readable probability, normalised to values between $[0, 1]$.

The probably most important mechanism is the *Self-Attention* mechanism. The core idea is that the input or output sequence pays attention to itself. This allows for capturing how different sequence parts refer to each other within the sequence. It is based on the notion of *Attention* which was introduced to allow for the selection of the hidden state with the most relevant information while generation (decoding). Summarising, *attention* captures the relation between the individual parts in the input and output sequences, respectively, and *Self-Attention* captures the relation between the individual parts in the input sequence itself.

Two more major architectures were derived from the initial notion of the Transformer architecture.
Causal Language Models are based on a decoder-only architecture. Being of autoregressive nature, they allow for great generation capabilities and can learn new things fast requiring only little data. They are typically used for next-token prediction, generating an output sequence step by step. A token in this sense is a word, symbol or part of a word, depending

on the tokenisation strategy. Popular examples are *GPT* by OpenAI, *Llama* by Meta, *Gemini* by Google and, very recently released, *Mistral 7B* by MistralAI.

Masked Language Models on the contrary are based on an encoder-only architecture, benefiting strong textual understanding but poor generation capabilities. This strong textual understanding is based on the capability of considering the surrounding context in a bidirectional way, taking into account past and future tokens when making predictions These models are typically used for text classification, sentiment analysis and named entity recognition. The most prominent representatives are Bidirectional Encoder Representations from Transformers (BERT) [15] and its descendants like RoBERTa, DistilBERT and ALBERT.

2.2. LLMs

2.2.1. Short Introduction

LLMs have emerged as cutting-edge Machine Learning (ML) systems ingesting huge amounts of data while pre-training and being capable of processing and generating text approximating human-level performance on certain benchmarks [16]. New developments in Artificial Intelligence (AI) research, as well as new milestones in hardware acceleration, created the foundation for the rapid development of LLMs. A pivotal milestone for LLM development was laid by Vaswani, Shazeer, Parmar, *et al.* [3] introducing the Transformer architecture. To train an LLM base, called a foundational model, large amounts of data (mostly from the internet) are ingested in a self-supervised setting. Most SOTA LLMs (e.g. GPT-3.5, GPT-4, Llama-2, etc.) are CLMs [16].
Even though other architectural approaches exist, Transformers are predominant in this domain, partly due to the use cases imposed upon them in the last couple of years.

2.2.2. Base Model

For this work, *Llama2* by Meta [4] was selected as the base model for fine-tuning, due to its recent release when starting this work, good resources and the fact that it is an open-source model.
Llama2 comes in three sizes, 3B, 13B and 70B and two flavours, the basic pre-trained *Llama2* and the fine-tuned *Llama2 chat* version. The pre-training was conducted on two trillion tokens, derived from publicly available data, and evaluated on seven different academic benchmarks as well as one safety benchmark. To be more specific, the huggingface version of Llama2 chat, *Llama-2-13b-chat-hf*[1], available on *huggingface.co* was utilised for this work's baseline and prototype fine-tuning.
Regarding model size, the 13B parameter chat-optimised version was chosen, due to preliminary experiments on small datasets showing more suitable responses. Additionally, given the nature of the used dataset, fine-tuning the non-chat version resulted in cryptic responses combining symbols, numbers and letters, not useful for further proceeding.

[1] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf, accessed 09.11.2023 21:50

Llama2 comes with its tokeniser which is a *Byte Pair Encoding* algorithm based on sentencepiece[2], a popular open-source text tokeniser and de-tokeniser developed by Google. Most importantly, *Byte Pair Encoding* allows for combining tokens that encode single characters and tokens that encode whole words.
Llama2's tokeniser contains a vocabulary of 32000 tokens with respecting token IDs alongside special tokens for sequence beginning (*bos_token* <s>, corresponding to 1), ending (*eos_token* <\s>, corresponding to 2), words or symbols the model does not know (*unk_token* <unk>, corresponding to 0) and the padding token, used to pad a sequence to guarantee same length among all sequences. Additionally, there are some special symbols dedicated to prompts: `<<SYS>>` and `<</SYS>>` for system messages and `[INST]`, `[/INST]` for instructions. The maximum number of tokens ingestible by *Llama2* is 4096.

Figure 2.3 illustrates the process of how Meta fine-tuned *LLama2*. The first step consisted of self-supervised pre-training of *Llama2* on publicly available data. Then, fine-tuning was conducted similarly as OpenAI did for *ChatGPT* by utilising Reinforcement Learning by Human Feedback (RLHF). The dominant language of the data in pre-training was English with a share of 89.70%. The pre-training process of *Llama2* took 368,640 GPU hours, plus the time for the fine-tuning process to create the *chat* version. [4]. For the rest of this work *Llama2 Chat* (13B) is simply referred to as *Llama2*.

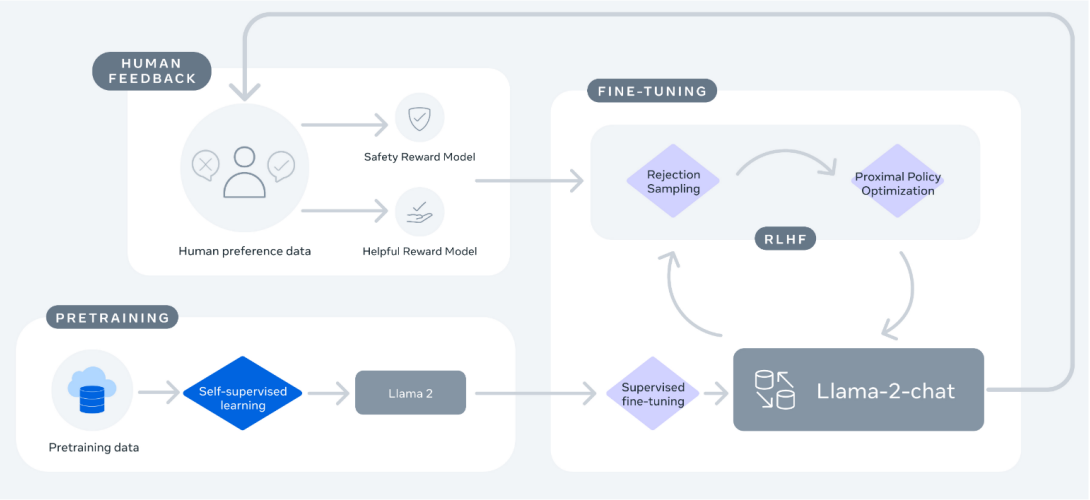

Figure 2.3.: Original pre-training and fine-tuning process of *Llama2* by Touvron, Martin, and Stone [4].

[2]https://github.com/google/sentencepiece, accessed 21.11.2023 19:12

2.2.3. Fine-Tuning LLMs

The process of adapting a pre-trained foundational model to a specific task is called fine-tuning. It allows a LLM to be streamlined on a certain skill, for example being able to respond in a human chat-like manner. This process can be performed with a variety of strategies:

Transfer Learning, *Instruction-Tuning*, *Alignment-Tuning* and *Parameter-Efficient Tuning* (PEFT). While all approaches have some similarities, PEFT methods are the most prominent approach, allowing to fine-tune LLMs on a single A100 80GB GPU or even on consumer-grade 24GB GPUs like the Nvidia RTX4090, depending on the base model size. There are three sub-strategies within PEFT:

- *Prefix Tuning* adds prefix vectors to all layers of the model, which are then fine-tuned, while the non-prefix part stays frozen [17].
- *Prompt Tuning* can be seen as a simplified version of *Prefix Tuning*. Prompting in the context of fine-tuning is defined as adding additional information to the input X the model can condition on, to maximise the likelihood of the correct Y. Usually, these prompts P are prepended to X parameterised by θ (model weights). Lester, Al-Rfou, and Constant [18] removed this restriction and introduced θ_P, a set of weights dedicated to the prompt tokens. While fine-tuning, gradient updates are only applied to θ_P via backpropagation while θ is kept frozen.
- *Adapter Tuning* introduced by Houlsby, Giurgiu, Jastrzebski, *et al.* [19] adds an encoder-decoder module either after or parallel to the attention and feed-forward layer, to reduce the parameters of the network by compressing the inputs into a lower dimension (bottleneck architecture). Nevertheless, Hu, Shen, Wallis, *et al.* [5] pointed out that adapter layers introduce latency to the model since the injected adapters are processed sequentially. Hence, they proposed the nowadays very prominent approach of Low-Rank Adaptation (LoRA).

2.2.4. LoRA

LoRA utilises a reparameterisation trick and builds up on the findings of Aghajanyan, Zettlemoyer, and Gupta [20], who pointed out that "pre-trained language models have low *intrinsic dimensions*" [20]. The intrinsic dimension is defined as the "minimal number of coordinates which are necessary to describe its points without significant information loss" [21]. These findings led Hu, Shen, Wallis, *et al.* [5] to the hypothesis, that the weight-updates analogously have a low intrinsic dimension or *intrinsic rank*, allowing for good performance by optimising much smaller decomposition matrices, since only decomposition matrices are optimised instead of the full weight matrix. Upon this hypothesis, Hu, Shen, Wallis, *et al.* [5] proposed their approach for LoRA, consisting of low-rank decomposition of the weight matrix $W_0 x \in \mathbb{R}^{d \times k}$ constraining the weights updates. Their approach is denoted as follows:

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{2.1}$$

whereas x denotes the input, h the reconstructed model, and $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$ the decomposition matrices, with $r \ll min(d, k)$. The idea was based on the fact that a full rank matrix $W_0 x \in \mathbb{R}^{d \times k}$ of rank r, always has a factorisation $W = W_a \times W_b$ with $W_a \in \mathbb{R}^{d \times r}$ and $W_b \in \mathbb{R}^{r \times k}$. The reparameterisation process can be seen in Figure 2.4. The authors used a random Gaussian initialisation for A and zero for B, so $\Delta W = BA$ equals zero at the beginning, which is scaled by $\Delta W x$ with $\frac{\alpha}{r}$, where α is a constant in r similar to a learning rate.
Summarised, the two decomposition matrices A and B, which are smaller in dimension due to r, create a low-rank approximation of the original model. Those decomposed matrices are then fine-tuned on the specific dataset before reconstructing the full model again. To retrieve the decomposition matrices, usually a Singular Value Decomposition [22] (or some adaptation of it) is used. With this approach, the number of parameters from the original matrix $W_0 x$ with $params_{base} = d \times k$ is reduced to $params_{LoRA} = r \times (d + k)$. According to the experiments of Hu, Shen, Wallis, *et al.* [5], where GPT-3 with 175 billion parameters was fine-tuned, the VRAM usage was reduced by up to 2/3 if $r \ll d$, the checkpoint size was reduced by $\times 10.000$ and a speed-up of 25% achieved.

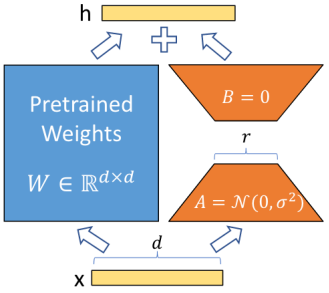

Figure 2.4.: LoRA reparameterization as proposed by Hu, Shen, Wallis, *et al.* [5].

For the sake of *Alignment Tuning* RLHF is a very popular methodology utilised by famous representatives such as *ChatGPT* [3] or *Llama2* [4]. The RLHF approach requires the training of a supervised reward model based on the feedback of human annotators (in terms of ranking) given multiple outputs. The trained reward model calculates a scalar value (reward) reflecting the quality of the result given a prompt and the model's output. In the next step, a chosen pre-trained model is fine-tuned using the reward model and an RL algorithm, of which most commonly the Proximity Policy Optimisation (PPO) is implemented [23]. In the context of RL, a policy is a strategy the model's agent can choose actions from based on the current state which achieves the highest cumulative reward. A policy is defined as the distribution of a finite set of actions a given a state s called trajectories $\tau = (a_1, s_1, ..., a_t, s_t)$, parameterised by θ (e.g. model weights) denoted

[3] https://openai.com/blog/*ChatGPT*, accessed 21.11.2023 13:00

as

$$\pi_\theta = P(a|s, \theta). \tag{2.2}$$

Even though the generated results are superb, Naveed, Khan, Qiu, *et al.* [16] point out that RLHF is complex, memory-intensive and unstable, making the process slow and costly. In their work, they mention a semi-automated process emerging in literature called Reinforcement Learning from AI Feedback [24], where specialised LLMs are exploited to imitate human feedback for dataset creation. This approach could reduce the efforts inflicted with RLHF, making it more adaptable even for smaller fine-tuning projects.

Further optimisation methods include quantisation strategies for hardware efficiency and mixed-precision tuning. Quantisation strategies tackle the issue of the rate of LLMs growing in size being higher than the growth in hardware capabilities [25]. As a solution, they compress the model weights to low-precision floating points or integer representation (e.g. FP32 to FP16).
Mixed-precision tuning introduced the idea of combining the use of different numerical precision methods while training. The core idea is to use *half-precision* (FP16) floating point representation while keeping the model accuracy of *single-precision* (FP32). This is achieved by storing weights, activations and gradients as FP16, but maintaining and updating a master-weights copy in FP32 during the optimiser step. Utilising *half-precision* (FP16) reduces the required memory and inference time of a NN, two of the core resources to consider in ML. The mixed-precision approach reduces the overall memory consumption of NNs by roughly 50% [26].

Lastly, it is worth mentioning the recent work of Dettmers, Pagnoni, Holtzman, *et al.* [27] who introduced QLoRA, combining LoRA and the concept of quantisation in proposing a new 4-bit float datatype, enabling the fine-tuning of a 65B parameter model on less than 48GB of VRAM memory.

2.2.5. Prompting

Prompting, in the context of LLM inference, is the common method used to retrieve a response by an LLM given a query. As opposed to fine-tuning, no weight updates are made to model adaptation. Hence, different strategies can be applied, yielding different results [16].
Most important for this work is the notion of *Zero-Shot Learning* and *In-Context Learning*. There are different approaches for the inference of an LLM. *Zero-shot* prompting refers to one of them, where the prompt for the model does not contain any exemplary item with the corresponding correct response. Whereas when utilising *In-Context Learning*, which is also known as *Few-Shot* prompting, the model's prompt is combined with one or multiple example input-output pairs to give the model some guidelines for the desired answer generation [16].

2.2.6. Drawbacks on LLMs

As for any technology, there are drawbacks to implementing LLMs. Especially important to consider in a medical context. Thirunavukarasu, Ting, Elangovan, *et al.* [28] pointed out seven of those issues in their work.

Recency states a big issue in current LLM development. Many models (e.g. GPT3.5, *GPT4*) were only trained on data up to a certain point[4]. This leads to the discussion, of whether the generated content is accurate and useful, which is of high importance in a clinical context.

Accuracy relates to a model's domain-specific knowledge. A prominent issue described as "*garbage in, garbage out*" is historically known in ML and states an important issue of LLMs being trained mostly self-supervised on data from the internet, with little to no mechanisms to check for the validity and accuracy of these inputs.

Coherence of LLMs is different to humans since the models do not understand language as we do. For LLMs or any NLP models, language is learned by statistical associations between words, predicting the most probable word. This leads to issues like *hallucination* "where inaccurate information is inverted and espoused lucidly" [28].

Transparency and Interpretability became topics at the latest since DNNs started to dominate the world of ML. Being complex in terms of hundreds of billions of neurons and a couple or more hidden layers drew the notion of *black box* to DNNs, making it hard to build trust in the decision-making process. A dedicated research stream of Explainable AI developed, developing different methods on how to achieve a better look into the *black box* of a model's decision-making process, becoming ever so important in a medical context. Yet, as concluded by Thirunavukarasu, Ting, Elangovan, *et al.* [28] these methods might be great in improving interpretability, but not sufficient enough to truly create trust [29].

Ethical concerns arise with any new technology but imposing an even wider discussion in the context of generative AI like LLMs. Biased models due to biased data, and adversarial prompts jailbreaking models to create hazardous answers paired with the strong text generation capabilities of such models, indistinguishable from human-written ones, impose many ethical considerations, requiring discussion and clear responsibilities.

Security and Privacy concerns are serious issues, especially in the clinical context, where patient and therapy information is sensible, limiting the deployment possibilities of LLMs such as *ChatGPT*, *GPT4*, *Gemini* and others. Besides, due to non-publicly available weights and the huge sizes, local deployment of such models becomes impossible.

Validation is mentioned as the seventh issue. LLMs should be evaluated in randomised controlled trials evaluating the effects of utilisation, as stated by the authors. The problem highlighted is the question about the way of benchmarking these results, imposing a problem that needs to be tackled.

[4] https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo, accessed 27.02.2024 21:01

2.2.7. Terminology

The terminology around LLMs gets mixed up a lot. For example, the term *Prompt Tuning* is used for a method in fine-tuning as well as a method in prompting the model for inference, even though being completely different. *Instruction Tuning* is utilised as its own method, as is PEFT by Naveed, Khan, Qiu, *et al.* [16]. Nevertheless, LoRA is sometimes described as part of the PEFT stack and sometimes not.
This work was conducted in a supervised instruction-based manner. This said, the model's input items always contained an instruction alongside the domain-specific data as well as the masked output label. Additionally, LoRA was utilised as the PEFT method, to reduce the memory load and enable single-GPU fine-tuning.

2.3. Relevance Ranking in Information Retrieval

IR systems are used to rank a set of documents by their relevance (or usefulness) given a certain search query. This is achieved by assigning a numeric score to each document and ranking descending from highest to lowest score.

2.3.1. Classical Approaches

Classical IR systems can be divided into three groups, *Vector Space Models*, *Probabilistic Models* and *Inference Network Models* [30].

Vector Space Models create high-dimensional vectors for documents and queries in which, if a term appears in the document, the corresponding position in the text vector is set to a non-zero value. A term can be defined as a word, keyword or phrase. The same is done for the query, allowing similarity calculation of the resulting vectors with e.g. cosine similarity [30], which simply measures the spatial closeness of two vectors.

Probabilistic Models first proposed by Maron and Kuhns [31], estimate the relevance probability of a document D given a query Q, which can be expressed with $P(R|D,Q)$. To achieve a ranking instead of a set of probabilities, the log odds are calculated by mapping the probability value from $(0,1)$ to $(-\infty,+\infty)$. The log-odds are also called Relevance Status Values [32] and are ranked descending based on the resulting score.

Last but not least, *Inference Network Models* are types of probabilistic models with the difference, that the document retrieval is based on a network representation and modelled as an inference process. The basic model consists of a document network and a query network, which can only have two values, *true* or *false*. The mechanism behind the *Inference Network Model* is based on the *Bayesian Network* [33].

2.3.2. Neural Network-based Information Retrieval

Over time a new research stream in IR emerged known as Learning to Rank (LtR) introducing NNs to IR, where the traditional models are integrated as features, delivering valuable information to model training. There are three main LtR approaches: 1) The

point-wise, 2) the pair-wise and 3) the list-wise approach. The main characteristics for all three types of LtR are 1) they are *feature based* approaches since a feature vector is created for query-document pair, reflecting the document frequency given a query, and 2) they are trained in an *discriminative* manner since there is always a specified *input space*, *output space*, *hypothesis space* and a *loss function* (with small differences based on the chosen approach) [34]. The hypothesis space is also called the *scoring function*, which calculates a relevancy score for a given input.

The point-wise approach approximates the LtR problem by a regression problem, given the training data has a numerical or ordinal score. Here, the input space contains the feature vector (input space, e.g. derived from the traditional models). The point-wise approach does not consider interdependencies between documents or the knowledge of some documents being associated with the same query. Hence, the final ranked list is solely ranked by the result of the scoring function, imposing limitations for most IR applications.

The pair-wise approach, on the other hand, is built on the binary classification of object pairs into correctly and incorrectly ranked classes. These pair-wise references are reflected in the output space, which tells what document is more relevant in a given document pair. The loss function is defined as measuring the predicted relative order between a document pair and its ground truth.

The list-wise approach [35] includes all documents associated with a query q in its input space. The ranking is achieved with a compound function, first giving a score to each document and then sorting these documents in descending order. Compared to the point-wise and the pair-wise approach, the advantage of the list-wise approach lies in the loss function considering the document positions within the ranked list of all documents.

Yet, with the establishment of SOTA performances achieved on various tasks with NNs, researchers started to experiment with NN architectures for IR. A new category of *Neural Information Retrieval* approaches emerged. These kinds of models create embedding representations for query and document to calculate the similarity based on metrics like cosine similarity or similar [36].

Recently, the Retrieval-Augmented Generation (RAG) architecture emerged as a very promising approach for the IR domain, achieving SOTA results on retrieval tasks and bypassing the usually static nature of traditional NN systems. RAG models are composed of two main components: 1) The retrieval component is based on a Dense Passage Retriever (DPR), and is used to calculate the top k documents z with the highest prior probability. For the calculation of this prior, a $BERT_{base}$ document encoder is utilised, generating a dense representation of a document $d(z)$ and a query $q(z)$. This calculation is called a Maximum Inner Product Problem (MIPS). 2) A generator component which can be modelled with any encoder-decoder architecture [37].

3. Related Work

This chapter provides a literature review of relevant publications. Section 3.1 tackles literature regarding LLMs in the medical domain, followed by publications about automatic patient-trial matching in Section 3.2. Section 3.3 combines both domains and summarises the most relevant work conducted in the domain of LLMs for patient-trial matching.

3.1. LLMs in Medical Application

Generative AI has shown great results in medical tasks. No wonder, that the healthcare sector was the sector with the highest investments in AI in 2022 [38]. Especially LLMs created new directions generating thousands of new possibilities for health care applications such as virtual health assistants, medical literature summarisation, conversational agents for patients as well as physicians, information extraction from EHR, public health surveillance in analysing news and relevant publications, etc.

A good example is*Med-PaLM 2*, which was released last year. *Med-PaLM 2* is a version of *PaLM 2* fine-tuned on medical domain data increasing their performance by 19% compared to *Med-PaLM 1*, achieving SOTA results on five out of nine multiple-choice medical question sets and very promising results in the long-form answer segments (explanations/ reasoning) based on qualitative evaluation. Very recently released and closely related to the idea of *Med-PaLM 2* was a fine-tuned version of Llama2 called *Meditron* available in sizes 7B and 70B. It was trained on 41.8 billion tokens from four different datasets [39]. The 70B-model achieved 70% on the MedQA[1] dataset, compared to 86.5% of *Med-PaLM 2*, being significantly smaller consisting of 540B parameters [2]).

Li, Li, Zhang, *et al.* [40] proposed *ChatDoctor*, a fine-tuned version of *Llama* (7B), trained on real-world doctor-patient interactions, standing out due to its small size, its augmented retrieval feature, extracting relevant information from multiple external databases to allow the model understand new or unknown medical terms. Even though the authors did not mention it, one could notice the resemblance of the RAG approach proposed by Lewis, Perez, Piktus, *et al.* [37].
ChatDoctor performed coequal to *ChatGPT* in answer generation on questions from the iCliniq[3] database, calculating the BERTScore [41] of both model responses. The BERTScore is calculated as a sum of the cosine similarities between the response and the reference sentence output tokens.

Last but not least and worth mentioning is *MedAlpaca* proposed by Han, Adams, Papaioannou, *et al.* [42], a medical adaption of Llama2 (13B) building up on the work of Taori, Gulrajani, Zhang, *et al.* [43]. The model was evaluated on the United States Medical Licensing Examination, which is split into three steps, achieving scores of 0.473, 0.477

[1] https://paperswithcode.com/dataset/medqa-usmle, accessed 29.11.2023 23:49
[2] https://www.cnbc.com/2023/05/16/googles-palm-2-uses-nearly-five-times-more-text-data-than-predecessor.html, accessed 10.12.2023 22:47
[3] https://www.icliniq.com/, accessed 01.12.2023 14:16

and 0.602, respectively, which outperformed *ChatDoctor* almost by threefold scoring 0.187, 0.185 and 0.148 for the three steps respectively. Han, Adams, Papaioannou, *et al.* [42] The most relevant finding for this work's context was that utilising LoRA and 8-bit quantisation reduced the performance drastically, rendering a result worse than the baseline. Though the authors noted, that they did not conduct any extensive hyperparameter optimisation.

Closing this section, one can see that a lot of effort has been put into the development of medical assistants, especially since LLMs were introduced. Nevertheless, model sizes, used datasets, hyperparameters, and utilised optimisation techniques have a strong impact on the results and vary from case to case. Further, evaluation approaches differ strongly, since no overarching standard was established, which might be unsuitable for the new paradigm. This leads to researchers only displaying a selection of their metrics, probably based on their best-performing evaluation strategies.

3.2. Patient-Trial Matching

One can distinguish two types of approaches for patient-trial matching, the *Structure-then-Match* and the *End-to-End* strategy.
The *Structure-then-Match* as proposed by Yuan, Ryan, Ta, *et al.* [44] extracts and structures key entities and relations from the EC in the first step, utilising techniques like Named Entity Recognition (NER), boolean logic and heuristic methods. In the second step, the structured information is translated into a query to retrieve relevant documents. The *End-to-End* approach on the other hand is an encoding-based approach, utilising NNs to create embeddings from the given information, which are then utilised for matching patient descriptions to ECs [45]. Since the *End-to-End* approach is the more relevant approach for this work, only related work utilising this approach is considered. In the following, the most promising approaches by participating teams from the TREC Clinical Trials track 2021 and 2022 are summarised, as well as some TREC non-related publications, achieving decent results. The TREC Clinical Trials track, which firstly stated the patient-trial matching task in 2021, introduced the *trec_eval* script in September 2016[4], which became a standard tool for the evaluation of any of the TREC tracks. The TREC is an annually occurring conference that has stated different challenges in the domain of IR for over 33 years by now. It is important to highlight, that in the case of the TREC Clinical Trials track, the teams always provide two results. Multiple automatic runs and multiple manual runs, of which the latter is based on a Human-in-the-Loop active learning strategy.

The team of Pradeep, Li, Wang, *et al.* [46] (h2oloo) achieved the best-performing results in the TREC Clinical Trials track 2021 as well as 2022, even though there is no publication for 2022 available. Building upon the insights of Koopman and Zuccon [47], who found out that short keyword-based queries are more effective than lengthy sentences, Pradeep, Li, Wang, *et al.* [46] proposed the *Neural Query Synthesis* method, which leverages the doc2query-t5 [5] model to generate multiple sentence-long queries from the given topics.

[4] https://github.com/usnistgov/trec$_e$*val*, *accessed*16.01.202413 : 25
[5] https://huggingface.co/doc2query, accessed 07.11.2023 22:43

After generating the queries, the team issued the queries one by one utilising two ranking algorithms, Best Match 25 (BM25) and Relevance Model 3 (RM3). All query results were then fused via Reciprocal Rank Fusion as proposed by Cormack, Clarke, and Buettcher [48] to create a first-stage candidate list.
In the second-stage the authors fine-tuned Med-Mono-T5 ($monoT5_{MED}$) [49], a neural Seq2Seq ranking model. They utilised the curated CT-to-Patient matching dataset by Koopman and Zuccon [47], containing 60 topics and 6,000 relevancy annotations to achieve their second-stage ranking model. The foundation for Pradeep, Nogueira, and Lin [49] work was laid by Nogueira, Jiang, and Lin [50], who proposed an adaption of the good text-generation capabilities of the T5 model [51] to the domain of document relevancy ranking. The basic idea is to use the model for generating a "true" or "false" label for relevant or non-relevant documents, respectively. Then, the probability for the respective label is computed via the $softmax$ function. This relevancy ranking is achieved in a point-wise ranking manner, utilising a template in the format:

$$Query\colon q\ Document\colon d\ Relevant\colon \{TRUE|FALSE\} \tag{3.1}$$

While fine-tuning Pradeep, Nogueira, and Lin [49] faced two problems, which they called the *training problem* and the *inference problem*. Since the eligibility and description fields of the CTs are lengthy multi-sentence fields, the authors faced issues in terms of the model's maximum number of input tokens, as well as the high computational costs. Hence, for the $Document\colon d$ part of the template, the eligibility and description fields were segmented using a defined window size. These segments were used as input to the $monoT5_{MED}$ base model to retrieve the highest-scoring segments, which were then utilised for fine-tuning. In total, three query templates were used to craft the training data, considering title, condition, eligibility and description. The first captured title, condition and eligibility. The second captured title, condition and description, and the third all of them.
Inference of their resulting $monoT5_{CT}$ model was conducted similarly. They segmented the lengthy eligibility and description fields through a defined window size and retrieved the highest scores of the three utilised templates. With *MaxP* a single score was retrieved. $monoT5_{CT}$ achieved a normalised Discounted Cumulative Gain @k (nDCG@k) nDCG@10 of 0.7118 and a MRR of 0.816.

The Alibaba DAMO team [52] proposed *TrialMatcher*, a first-stage retriever architecture containing two separate encoders, one for the patient notes and one for the CT. The authors initialised both encoders based on ClinicalBERT [53] and pre-trained them on all available CT samples provided by the TREC. The pre-training was conducted in a contrastive learning manner, where the model's task was to match patient-to-trial as well as trial-to-patient. Jin, Tan, Zhao, *et al.* [52] chose the triplet loss as a loss function described with

$$\mathcal{L}(A,P,N) = max(\|f(A) - f(P)\|_2 - \|f(A) - f(N)\|_2 + \alpha, 0). \tag{3.2}$$

The final ranking for the second stage was based on an Embedding Based Retrieval approach, where the embedding space distance between patient notes and CT is used for the ranking. To realise the second stage re-ranking, Jin, Tan, Zhao, *et al.* [52] fine-tuned a clinicalBERT model on the dataset curated by Koopman and Zuccon [47]. With their best run (*damoebrtog*) they achieved a nDCG@10 of 0.595 and a MRR of 0.608, placing second in the TREC Clinical Trials track 2021. The authors concluded that solely Embedding Based Retrieval approaches are useful but not sufficient. Further, they argue that a large noisy set of instances could be more suitable for fine-tuning the second stage re-ranker than a small, clean one.

In the pivotal work of Gao, Xiao, Glass, *et al.* [54], the authors presented COMPOSE: A patient-trial matching system achieving an accuracy of 0.837 on the trial-level classification. Gao, Xiao, Glass, *et al.* [54] used a pseudo-siamese network architecture with two branches using similar and dissimilar pairs for similarity learning. A CNN utilising highway layers [55] was trained on learning EC embeddings accompanied by a memory network [56] for the patient's EHR.
The main idea is to capture both modalities, EC and EHR, into a shared latent space. One of the specialities of this work is the utilised EHR memory network, consisting of three sub-memory networks (diagnose, procedure and medications) capturing a patient's longitudinal data of physician visits, creating an accessible patient history. Every sub-memory network is composed of "four memory slots to store information from fine-grained to coarse levels" [54]. These four levels create a hierarchical network, leveraging medical taxonomy, ranging from broad terms to specific ones. This allows the EC network to calculate its embedding distances with every entry in the memory network individually. The motivation for such an approach rose from the vast landscape of medical terminology. To predict if a criteria e matched a certain patient, the authors computed a *best matching memory* $\tilde{m}$ value. Given a criterion e (inclusion or exclusion), they calculated the attention weights for each sub-memory within the memory network and summed them up to retrieve $\tilde{m}$. In the last step, a Multi-Layer Perceptron was used to create an additional embedding m_d for the patient's demographics, which was then concatenated to $\tilde{m}$ before using the *Softmax* function to predict $\hat{y}$.
For loss computation, Gao, Xiao, Glass, *et al.* [54] proposed a composite loss function consisting of two parts, the classification loss $\mathcal{L}_c$ and the inclusion/exclusion loss $\mathcal{L}_d$. For the classification loss, cross-entropy was used. For the inclusion/exclusion loss, the authors separated the EC in inclusion and exclusion criteria, before computing the distance between $\tilde{m}_I$ or $\tilde{m}_E$ with $d(e, \tilde{m}_I)$ or $d(e, \tilde{m}_D)$, respectively. The inclusion/ exclusion loss $\mathcal{L}_d$ is minimised for the inclusion criteria and maximised for the exclusion criteria. The final loss $\mathcal{L}$ was achieved in adding up $\mathcal{L}_c + \mathcal{L}_d$.
COMPOSE is the successor of DeepEnroll by Zhang, Xiao, Glass, *et al.* [57] since it was built on a very similar architecture, whereas the biggest difference is, that DeepEnroll does not focus on the separate contemplation of inclusion and exclusion criteria.

Recent work by Theodorou, Xiao, and Sun [58] builds up on the ideas of COMPOSE

by providing more interpretable results and slightly higher accuracy on the trial- and criteria-level classification tasks 0.849 and 0.956, respectively. Theodorou, Xiao, and Sun [58] proposed TREEMENT, a network differing from COMPOSE mainly by introducing a dynamic tree-based patient EHR memory network. For each patient, an individual memory tree is created. The memory network itself is implemented with the same attributes as in COMPOSE: Three sub-memory networks which hold four memory slots each to store information. Besides higher accuracy values in classification, TREEMENT introduced two major benefits: 1) The slightly adjusted architecture provides a more efficient memory network reducing the number of parameters from 497,280 (COMPOSE) to 199,299, and 2) allowing for an easier and more narrow interpretation of the predictions due to the the tree-structured memory-network.

Summarising, the common approach for relevancy retrieval in patient-trial matching (and other IR tasks) can be split into two steps: 1) An initial retrieval step to retrieve N documents using well-established retrieval algorithms (e.g. BM25), and 2) the re-ranking step, rearranging the initially retrieved documents based on more elaborate architectures like NNs.
Unfortunately, a proper comparison between the results of the TREC Clinical Trials track and the TREC non-related systems is not possible, since different metrics were applied as well as different datasets for the patient EHRs utilised. Nevertheless, the contributions of the last years highlight the importance of the patient trial-matching task and show good progress in the development of systems applicable in real-world settings, which might help clinicians in patient-trial enrolment and further tasks very soon.

3.3. LLMs for Patient-Trial Inference

In a recent work of Jin, Wang, Floudas, *et al.* [59] TrialGPT, a GPT-3.5 model (185 billion parameters) fine-tuned via an *In-Context Learning* approach, was introduced. The model's task was to predict a patient's eligibility on the criteria-level classification, given the patient's medical note. For the training, the authors used different biomedical and clinical natural language inference datasets, which they did not specify further. For the final inference, the authors split inclusion and exclusion criteria into two sets of data. The task of excluding ineligible clinical trials was modelled as a binary classification task. The model was tested in two ways. On the one hand, 415 manually evaluated patient-criterion pairs were evaluated qualitatively, showing the model's explanation and localisation capabilities. On the other, in the eligibility classification of 184 patient descriptions and 18,238 CTs. The final trial-level eligibility prediction was achieved by aggregating the criteria-level predictions. Jin, Wang, Floudas, *et al.* [59] pointed out, that most errors made by the model were due to insufficient medical knowledge.
Nevertheless, their work shows the decent capabilities of foundational models like GPT-3.5 in such complex language tasks, achieving an explanation and eligibility accuracy of 0.850, respectively. Further, TrialGPT achieves an nDCG@10 of 0.748 with optimal feature combination.

Hamer, Schoor, Polak, *et al.* [60] proposed a patient pre-screening approach by leveraging the capabilities of *ChatGPT* in assisting a physician at work. They used prompting strategies (one-shot, selection-inference, chain-of-thought) to first, prompt the model to check criteria-by-criteria which criteria are screenable and second, prompt the model if the screenable criteria are met or not and give a rationale on the reasons why. In their third and last step, a Physician-in-the-Loop was manually evaluating the retrieved model responses. With this approach Hamer, Schoor, Polak, *et al.* [60] achieved an overall accuracy of 0.720 at criteria-level and a precision of 0.710 at trial-level classification. A recall of 0.500 was achieved. They stated that the enhancement of the recall should be of major concern, to assure that no potential treatment is overlooked. Also, they concluded, that the Physician-in-the-Loop approach can not only prevent hallucination but also help in terms of interpretability of the final results.

Since Koopman and Zuccon [47] showed the dominance of queries over free text, the question came up, if LLMs can help in generating such queries from patient descriptions. This question was recently tackled by Peikos, Symeonidis, Kasela, *et al.* [61], where the researchers evaluated ChatGPTs capabilities of query generation based on a given patient note provided by the TREC Clinical Trials track 2021 and 2022 datasets for first-stage document retrieval. They experimented with different prompt sets and system architectures, combining different pre-processing steps (e.g. negation removing, term expansion) and compared their first-stage retrieval results to the best performing TREC Clinical Trials track 2021 and 2022 team (h2oloo). The authors achieved a nDCG@10 of 0.512 and a MRR of 0.623 with their two best approaches (IEMT, NRIEMT), which beat the first-stage retrieval runs of h2oloo in both years.
After evaluation, in comparing the LLM-generated queries to human-generated queries, Peikos, Symeonidis, Kasela, *et al.* [61] concluded that "*ChatGPT* might be better for information extraction for this task compared to medical experts, under certain circumstances" [61]. Further, the authors were concerned regarding the lack of knowledge, about whether *ChatGPT* was exposed to the TREC 2021 datasets since there is an overlap between the date upon which OpenAI used data for the training of *ChatGPT* and the publication date of the TREC 2021 datasets. Further, small variations in the output, even though the temperature value was set to 0 were noticed and the overall black-boxness of the *ChatGPT* API, which might affect the outcomes of the experiments, was criticised. However, they experienced robustness in running the experiment two times achieving the same results, under the limitation of using the same account. Even more concerns were imposed upon privacy and security topics. Hence, the authors appealed for de-identification procedures before using *ChatGPT* on real patient data.

With a similar idea Yuan, Tang, Jiang, *et al.* [62] leveraged *ChatGPT* for augmenting CTs EC before feeding those into the embedding-based SOTA patient-trial matcher COMPOSE. The EC augmentation approach resulted in an enhancement of the model's precision, recall and F1 scores on the patient-trial level by 8.6% (0.715 to 0.801), 8.2% (0.748 to 0.830) and 8,4% (0.731 to 0.815), respectively.

An even more recent work by Wong, Zhang, Gu, *et al.* [45] elaborated on the *In-Context*

Learning [63] capabilities of cutting-edge LLMs, such as GPT-4, for end-to-end CT matching. Their experiment showed that GPT-4 delivers reasonable results with no more than three examples, already outperforming some baseline models like Criteria2Query [44] in extracting inclusion and exclusion criteria (structuring task). Wong, Zhang, Gu, *et al.* [45] limited their work on the entity extraction of EC in oncology as a case study. Therefore, the model was directed to retrieve histology and biomarker inclusion and exclusion criteria, achieving an F1 score of 65.4 for histology and 72.5 for biomarkers, respectively. They highlighted key growth areas required for enhancing the end-to-end CT matching which are 1) addressing context length limitations and 2) more elaborate approaches in criteria extraction and structuring. Further, they argued that GPT-4 is not suitable for checking all patient-trial pairs, but is only useful for a higher-quality re-ranking step on several already well-received candidates.

4. Patient-Trial Classification

This chapter gives an overview of the patient-trial classification task introducing this work's prototype. Therefore, Section 4.1 explains the used datasets, including all required preprocessing steps, continuing with the introduction of this work's prototype architecture in Section 4.2. The evaluation strategy is described in Section 4.3, followed by an in-depth review of the prototype's fine-tuning procedure and its evaluation details in Section 4.4. Lastly, the hardware setup is described in Section 4.5, continued by Section 4.6 discussing the findings and limitations of the proposed approach.

4.1. Datasets

Two publicly available datasets are used: 1) the Dataset provided for TREC Precision Medicine Track 2021[1] and 2) the dataset provided for TREC Precision Medicine Track 2022[2]. Both datasets contain the same snapshot of the *clinicaltrials.gov* registry from April 2021, composed of 375,581 CT in XML format and 125 patient descriptions. They are composed of 75 distinct patient descriptions released in 2021 and an additional 50 released in 2022. Additionally, gold label files for both years were published after the conferences took place containing a total of 59.133 judged patient-trial pairs [9]. In the following, the individual parts of the datasets are shortly explained:

- CTs: Each clinical trial is assigned a National Clinical Trial (NCT) ID, for example "*NCT00392756*". CTs are provided in XML format, containing information such as title, summary, overall status, start date, study type, EC and many more. CTs are usually bound to a location as well to one of nine recruitment statuses such as *recruiting*, *active but not recruiting*, *completed* and *terminated*, just to name a few. However, the TREC Clinical Trials track task does not consider the location nor the status of the clinical trial, since it is not seen as a semantic aspect relevant to IR [9].
- Patient descriptions: Each patient description contains a summary of the patient, which resembles a doctor's letter. An example can be seen in Figure **??** This summary covers information about the patient's age, disease, past and ongoing therapies, medication, family medical history, health-induced limitations and any other information that might be relevant. Nevertheless, like in reality, the patient descriptions can also have missing aspects or non-informative ones. In the TREC Clinical Trials track the patient's description is referenced as patient topic. However, for this work, the term patient's description is used.
- Gold labels: The gold label files provide a mapping between the patient description, the NCTs and the numerical representation of the label. These files were created by the conference organisers in pooling examples from all runs of all participating teams. This pooled samples were judged in cooperation with the *Oregon Health and Science*

[1] https://www.trec-cds.org/2021.html, accessed 12.01.23 23:22
[2] https://www.trec-cds.org/2022.html, accessed 12.01.23 23:23

University and mapped to one of three possible labels: *eligible* (2), *not eligible* (1), and *not relevant* (0). In 2021 it contains 35,328 patient-trial pairs, with 5,570 (16%) *eligible*, 6,019 (17%) *excluded* and 23.739 (67%) *not relevant* items. In 2022 the gold label file contains 35,394 judged pairs, of which 3,949 (11%) are *eligible*, 3,047 (9%) *excluded* and 28.398 (80%) *not relevant*.

The data provided for the TREC Clinical Trials track 2023 challenge was not considered in this work, since the results were not published during this work's experimentation phase, and since a new data structure was introduced, not suitable in combination with the topics provided in 2021 and 2022.

```
<topics>
  ...
  <topic number="75">
    The patient is a 55-year-old man who was recently diagnosed with Parkinson's disease. He is complaining of slowness of movement and tremors. His disease is ranked
    as mild, Hoehn-Yahr Stage I. His past medical history is significant for hypertension and hypercholesterolemia. He lives with his wife.  They have three
    children.  He used to be active with gardening before his diagnosis. He complains of shaking and slow movement. He had difficulty entering through a door, as he
    was frozen and needed guidance to step in.  His handwriting is getting smaller. He is offered Levodopa and Trihexyphenidyl. He is an alert and cooperative man who
    does not have any signs of dementia. He does not smoke or use any illicit drugs.
  </topic>
</topics>
```

Figure 4.1.: Patient's description example from the TREC Clinical Trials track dataset.

4.1.1. Pre-processing

Dataset Item Crafting

In the first step, all required data sources are accessed and prepared in an Extract Transform and Load (ETL) pipeline to fit the defined dataset structure 4.2.

```
item = {
    "id": id_string,
    "topic_year": topic_year,
    "instruction": config[version]["instruction"],
    "topic": f"Here is the patient note:\n{cleaned_topic}",
    "clinical_trial": f"Here is the clinical trial:\n{ct_final_input}",
    "response": config[version]["response"],
    "output": f"{category}",
}
```

Figure 4.2.: The developed dataset item format for fine-tuning.

The gold label file containing the NCT ID, the topic number and the respecting label was used as a Single Source of Truth to extract relevant data from the different sources and create the first raw dataset, which is then split into training and testing datasets based on a set of patient description IDs and their corresponding year.

Only a few cleaning procedures were run on the patient descriptions like white space removal since the provided data had a high quality already. For the provided CT snapshots, on the other hand, more work was required. Firstly, neither all entries provided within the CTs are relevant, nor is *Llama2* capable of ingesting this much textual input. The CT was analysed for the most relevant data for extraction, cleaning and a final concatenation with labels and patient descriptions. This led to the extraction of the title, the CT summary and the criteria block, containing inclusion and exclusion criteria, guided by the template

of Pradeep, Li, Wang, *et al.* [46]. Additionally, to enhance the distinctiveness of the three labels for the fine-tuning and inference stages, the label "not eligible" was changed to "excluded" and the label "not relevant" to "irrelevant".

Even though *clinicaltrials.gov* provides a protocol for CT elements, which specifies that EC should be noted as a bulleted list with the respecting headers *Inclusion Criteria* and *Exclusion Criteria*, it was noticed that especially older CTs did not contain the criteria headers within their eligibility block. Since these two keywords were central for the pre-processing, to make the correct cut between inclusion and exclusion criteria, only those CTs consistent with the current *clinicaltrials.gov* CT protocol were considered for the final dataset. This way, it was possible to make sure, that all items have the same structure and the required keywords guiding the model's attention mechanism.

The cleaning process itself involved removing white spaces and line breaks, plus replacing double quotes with single quotes. Special characters were kept as far as possible, to not lose valuable information within the CT descriptions. Figure 4.3 shows a shortened CT in its raw XML format. Relevant information is extracted as shown in Table 4.1.

Title: Study on Newborn Babies With a Yellow Skin Color (Neonatal Jaundice Study)
Summary:Background: Neonatal hyperbilirubinemia is the most common reason for admission in the neonatal period (first month of life) worldwide and at SMRU. The skin of the newborn baby becomes jaundiced, which is caused by a high level of bilirubin in the blood. In some neonates, the level of bilirubin increases to a level that can cause brain damage or even death. There are different causes known that can lead to higher levels of bilirubin, for example, G6PD deficiency and prematurity. In the case of neonatal hyperbilirubinemia, the neonate needs to be treated with phototherapy (blue light therapy). If there is prolonged jaundice (>21 days), further investigations need to be done.
Inclusion Criteria:

1. Written or thumbprint informed consent from the mother during pregnancy.
2. Neonates who are born to mothers who followed antenatal care at SMRU antenatal clinics.
3. Neonates who are born in an SMRU clinic OR neonates who are born outside SMRU but visit an SMRU clinic within 48 hours after birth OR neonates who are born outside SMRU and present with neonatal jaundice at any moment in the first 8 days.

Exclusion Criteria:

1. No written or thumbprint informed consent from the mother during pregnancy.
2. Neonates who are born to mothers who did not follow antenatal care.
3. Neonates < 28 weeks gestation.
4. Neonate born outside SMRU and present > 48 hours after delivery without jaundice.

URL: `https://classic.clinicaltrials.gov/cthereasoningtask/show/NCT02361788`

Table 4.1.: Example of cleaned and filtered clinical trial.

```
<clinical_study>
  ...
  <brief_title>Point-of-Care Testing of Coagulation in Patients Treated With Direct Oral Anticoagulants 1</brief_title>
  ...
  <brief_summary>
    <textblock>
      The investigators study aims to test the correlation between the CoaguChek point-of-care&#xD;
      testing (POCT) device (Roche, Switzerland) and low plasma levels of all three currently&#xD;
      approved direct oral anticoagulants (DOAC; rivaroxaban, apixaban and dabigatran) and to&#xD;
      determine the diagnostic accuracy of POCT to rule out or detect relevant levels of DOAC in&#xD;
      real-life stroke patients.&#xD;
    </textblock>
  </brief_summary>
  <overall_status>Completed</overall_status>
  <start_date>May 2013</start_date>
  <completion_date type="Actual">September 2014</completion_date>
  <primary_completion_date type="Actual">August 2014</primary_completion_date>
  <study_type>Observational</study_type>
  ...
    <criteria>
      <textblock>
        Inclusion Criteria:&#xD;
&#xD;
          -  Stroke patients, who were newly started on oral anticoagulation with rivaroxaban,&#xD;
             apixaban or dabigatran for secondary prevention of thromboembolic events&#xD;
&#xD;
          -  Age ≥ 18 years&#xD;
&#xD;
        Exclusion Criteria:&#xD;
&#xD;
          -  Vitamin K antagonists or direct oral anticoagulants ≤ 14 days prior to study&#xD;
             participation&#xD;

          ...
      </textblock>
    </criteria>
    <gender>All</gender>
    <minimum_age>18 Years</minimum_age>
    <maximum_age>N/A</maximum_age>
    <healthy_volunteers>No</healthy_volunteers>
  </eligibility>
  ...
</clinical_study>
```

Figure 4.3.: A shortened example of a clinical trial in XML format.

Lastly, based on the defined item structure, the pre-processed CT, alongside the patient description, year, handcrafted instruction, response initiator, label and unique ID were added as new items to an array of objects one by one, until the processing of all relevant items was finished. This unique ID was created by concatenating a simple iteration counter, a patient's description ID, its corresponding year and the NCT ID of the CT.

Instruction Crafting

The output of an LLM is highly dependent on the prompt utilised. So important, that the term *prompt engineering* became its discipline and a new skill to learn, backed by resources like the *Prompt Engineering Guide* [64]. Hence, the input data had to be accompanied by an instruction.
For this work, 13 different instructions were developed and tested iteratively, ranging from very simple to more detailed. 12 instructions were streamlined for the classification task, of which the best was used as the foundation for the instruction adjusted for the reasoning task.

The best-performing instruction was inspired by the work of Jin, Wang, Floudas, *et al.* [59], which was further tweaked by using *ChatGPT*. This version was tested against the initial version by inferencing the model and checking which instruction creates responses closer to a) the label only for the classification task and b) step-by-step extraction with more human-like reasoning. Table 4.2 displays the best-performing instruction, enhanced by prompting *ChatGPT*. The desired output labels are always passed as part of the instruction. Given

an input X, the designed instruction forces the model to output one of the three possible labels $Y = \{A : eligible, B : excluded, C : irrelevant\}$.

Instruction: Hello. You are a valuable assistant for clinical trial recruitment. Your primary task is to meticulously evaluate a given patient note in comparison to the inclusion criteria of a clinical trial to determine the patient's eligibility. The patient's note serves as a concise medical summary that includes information about the patient's physical and mental condition, past medical procedures, current medications, complaints, and other pertinent details. Inclusion criteria are the specific factors that enable an individual to participate in a clinical study. These criteria are typically based on characteristics such as age, gender, the type and stage of a disease, previous treatment history, and other medical conditions. Conversely, exclusion criteria are the factors that disqualify someone from participating in a clinical study, often sharing similarities with inclusion criteria. To establish a patient's eligibility for a clinical trial, it is imperative that all inclusion criteria are satisfied. Simultaneously, none of the exclusion criteria should be met. This ensures that the patient is indeed eligible for the clinical trial. Inclusion criteria are listed after the "INCLUSION CRITERIA" keyword. Exclusion criteria are listed after the "EXCLUSION CRITERIA" keyword. Please respond by selecting one of the options in the bracket: A: eligible, B: excluded, C: irrelevant. **Response Initiator**: Is the patient eligible for the clinical trial?

Table 4.2.: Best performing of the 12 developed instructions, plus the used response initiator.

Train-Test split

To achieve a 90/10 train-test split, a set of patient descriptions, alongside their mapped CTs was removed from the dataset after initial pre-processing. The resulting training and testing datasets contain 55,120 and 5,902 items, respectively. Later on, before entering the training loop, a validation set is created by extracting 10% of the training data.
Preliminary experiments showed that adding a *one-shot* example allows for more concise results in the model's responses and a reduced amount of empty or non-evaluable items. Hence, a one-shot example based on the task at hand (classification or reasoning) was added to every item in the testing dataset during runtime. An example of a final dataset sample is shown in Table 4.4.

Dataset Filtering, Balancing

After the first coarse pre-processing steps and the splitting of the training and testing dataset, more delicate data filtering and balancing were conducted. This step is configurable via configuration files and hence, conducted after the initial dataset generation and dataset splitting. Further, this step was required due to the tested capacity limit of roughly 1000 words for fine-tuning on a 48GB GPU and the many items exceeding this word limit in the datasets created in the first pre-processing steps.
For inference, the upper limit was set to the maximum of 1500 words, corresponding to

```
"id": "35597_75-2021_NCT02902510",
"topic_year": 2021,
"instruction": "Hello. You are a valuable assistant for clinical trial recruitment. Your primary task is to meticulously evaluate a given patient note in
comparison to the inclusion criteria of a clinical trial to determine the patient's eligibility. The patient's note serves as a concise medical summary that
includes information about the patient's physical and mental condition, past medical procedures, current medications, complaints, and other pertinent details.
Inclusion criteria are the specific factors that enable an individual to participate in a clinical study. These criteria are typically based on characteristics
such as age, gender, the type and stage of a disease, previous treatment history, and other medical conditions. Conversely, exclusion criteria are the factors
that disqualify someone from participating in a clinical study, often sharing similarities with inclusion criteria. To establish a patient's eligibility for a
clinical trial, it is imperative that all inclusion criteria are satisfied. Simultaneously, none of the exclusion criteria should be met. This ensures that the
patient is indeed eligible for the clinical trial. Inclusion criteria are listed after the 'INCLUSION CRITERIA:' keyword. Exclusion criteria are listed after
the 'EXCLUSION CRITERIA:' keyword. Please give a final summary at the end and respond by selecting one of the options in the bracket: {A: eligible, B: excluded,
C: irrelevant}.",
"topic": "Here is the patient note:\nThe patient is a 55-year-old man who was recently diagnosed with Parkinson's disease. He is complaining of slowness of
movement and tremors. His disease is ranked as mild, Hoehn-Yahr Stage I. His past medical history is significant for hypertension and hypercholesterolemia. He
lives with his wife. They have three children. He used to be active with gardening before his diagnosis. He complains of shaking and slow movement. He had
difficulty entering through a door, as he was frozen and needed guidance to step in. His handwriting is getting smaller. He is offered Levodopa and
Trihexyphenidyl. He is an alert and cooperative man who does not have any signs of dementia. He does not smoke or use any illicit drugs.",
"clinical_trial": "Here is the clinical trial:\nTitle: Therapeutic Yoga to Improve Function in Parkinson's Disease\nSummary: This is a single blind, randomized,
wait-list controlled, phase II exploratory pilot study.\nIntervention Type: Other\nINCLUSION CRITERIA: - Diagnosis of Parkinson's Disease with a rating of 1.5-4
on the Modified Hoehn and Yahr Scale of Parkinson's Disease Progression - Endorsement of FoF,44 able to stand and walk 10 meters with or without an assistive
device - >18 years old; able to speak English - Score >4 out of 6 on the short Mini Mental Status Exam - Able and willing to attend twice weekly sessions for 8
weeks \nEXCLUSION CRITERIA: - People with self-reported life expectancy <12 months - Inability to attend sessions due to transportation issues - Current
involvement with other physical activity (including yoga), rehabilitation, or other intervention studies - Inability or refusal to provide informed consent",
"response": "Is the patient eligible for the clinical trial? Let's think step by step. ",
"output": "B: excluded"
```


Figure 4.4.: Example item from the testing dataset.

approximately 3072 tokens, covering most of the items in the testing dataset and giving the model enough tokens for a thorough response before hitting the model's limit of 4096 tokens. Hence, all entries exceeding the maximum word count of 1000 and 1500 words were removed from the final training and testing datasets, respectively.

Since the classes appeared to be very unbalanced, requiring adjustments. Data balancing was conducted on the topic level (micro balancing) with the goal of creating a roughly 50/50 (binary balancing) split between positive (eligible) and negative (excluded, irrelevant) labels. Therefore, the datasets were grouped by their unique topics and aggregated by the different labels.
With this balancing goal, the positive label had to appear twice as often as the individual negative labels. The aspired individual label count per topic was calculated with

$$\min_i \left(\sum_{j=1}^{n} x_{ij} \right) \tag{4.1}$$

Equation 4.1 provides the minimum aggregated label count among the three classes. Further, the aspired label count was divided by two for the positive label ($\frac{1}{2}$). The other half was split on the negative labels with ($\frac{1}{4}$, $\frac{1}{4}$). This way, an approximate 50/50 split was achieved assuring that positive and negative classes appear in an almost equal manner among each individual topic and all its permutations with the different CTs. After balancing, the final training dataset contained 11,348 items. Table 4.3 shows the class distribution of the final training dataset.

	Before F&B	After F&B
A: eligible	7460	4983
B: excluded	7272	3189
C: irrelevant	40388	3189
Total	55120	11361

Table 4.3.: Class distribution of training dataset before and after all pre-processing steps.

The same procedure was run for the testing dataset, except for the binary balancing, which was not conducted anymore to allow for a bigger sample pool resulting in 5,920 items. The

class distribution within the testing dataset is displayed in Table 4.4

	Before F&B	After F&B
A: eligible	960	931
B: excluded	659	621
C: irrelevant	4183	3984
Total	5802	5536

Table 4.4.: Class distribution testing dataset before and after all pre-processing steps.

4.2. Architectural Overview

Figure 4.5 shows all components included in this work's prototype. Based on the available TREC Clinical Trials track gold labels, a supervised learning approach was utilised for the fine-tuning of *Llama2*. The *llama-recepies*[3] by Meta was adapted for this work. The resulting codebase serves two purposes:

1) Providing a set of configuration-based scripts runnable via terminal making experimentation and fine-tuning of LLMs easy and trackable. 2) Allowing for configuration-based batch inference for experimentation with different instructions and hyperparameters, including the calculation of typical performance metrics tracked by a centralised tracking system.

The architecture consists of four major blocks, 1) the *pre-processing*block, containing the first non-configurable pre-processing steps, 2) the *fine-tuning* block, providing a configurable way to tweak the datasets and run fine-tuning with different settings defined in the fine-tuning configuration file, 3) the inference block, resembling the fine-tuning block but running the fine-tuned model in *generation* mode with the required inference hyperparameters and 4) the evaluation block, providing some response JSON post-processing scripts and functions for performance metrics calculation. All relevant inputs and outputs are tracked automatically with MLflow[4], a tool for tracking the development and for monitoring ML models in productions.

MLflow was developed to provide researchers and ML developers with an open-source tracking platform for the end-to-end ML lifecycle. MLflow flow allows the tracking of hyperparameters, training and inference metrics, and artifact versions via a model registry. Additionally, MLflow allows for a simplified model deployment and model evaluation, providing the user with a simple User Interface and customisable data plots. *Pytorch*[5] was utilised as python based end-to-end ML framework.

4.3. Evaluation Metrics

There are a couple of typical evaluation metrics from classification tasks and IR such as accuracy, precision, recall, F1 score, Precision @k (P@k) and nDCG@10 utilised in this work.

[3]https://github.com/facebookresearch/llama-recipes, accessed 20.08.23 10:35
[4]https://mlflow.org/, accessed 21.11.2023 19:40
[5]https://pytorch.org/, accessed 14.01.24 15:47

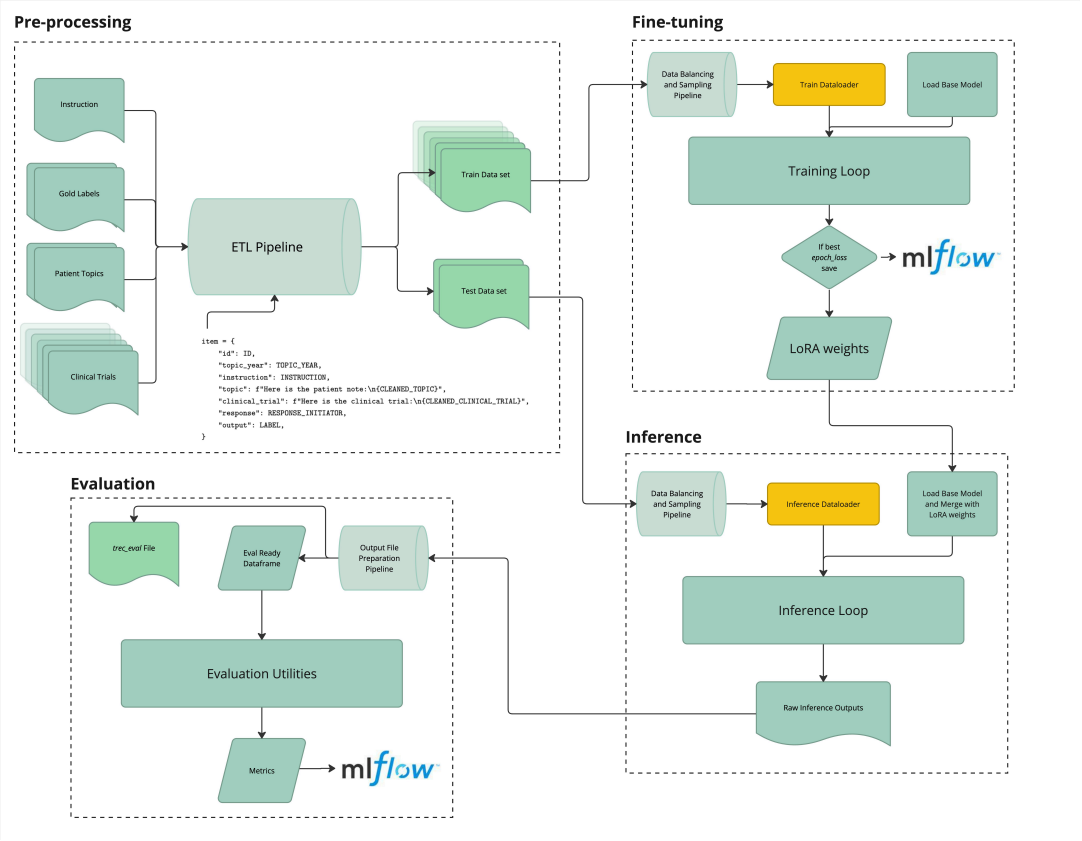

Figure 4.5.: The prototype's architecture is made of four major building blocks.

The **Accuracy** metric 4.2 is the most prominent metric found in ML. For binary classifications, it is defined as

$$accuracy = \frac{\text{TP} + \text{TN}}{\text{TP+TN} + \text{FP+FN}}, \tag{4.2}$$

whereas TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative. In the case of multi-class classification, accuracy is denoted as

$$accuracy = \frac{\text{Correct classifications}}{\text{All classifications}}. \tag{4.3}$$

Precision 4.4 and **Recall** 4.5 also play an important role in IR and ML in general. The precision score calculates the relative amount of relevant items within all retrieved items. The recall score (also called sensitivity) calculates the relevant retrieved items considering all relevant items.

$$precision = \frac{\text{TP}}{\text{TP+FP}} \tag{4.4}$$

$$recall = \frac{\text{TP}}{\text{TP+FN}} \tag{4.5}$$

A downside of the precision score comes up when the dataset is very unbalanced, which can result in high precision scores, even though the model's performance might be poor (e.g. only one relevant element among 100 samples, if retrieved by the model, equals a precision of 100%). The downside of the recall score is that it requires knowledge about

all relevant items, which especially in IR is rarely the case.

The **F1** score 4.6 represents the harmonic mean of the *precision* and *recall*. The F1 score is denoted as

$$F_1 = 2\frac{precision \cdot recall}{precision + recall} \tag{4.6}$$

representing *precision* and *recall* symmetrically. At the same time, this highlights the downside of **F1**, assuming *precision* and *recall* have the same importance.

Lastly, the Area Under the Receiving Operator Curve (**AUROC**) is a metric to tell the discrimination capabilities of a model between positive and negative cases. It compares the *True Positive Rate* with the *True Negative Rate*. In the case of a multi-label problem, two approaches are possible: 1) *One vs. Rest*, or 2) *One vs. One*. In the former case, a binary problem is created by separating one label and comparing it to the others. For the latter case, the problem is split into multiple binary problems, comparing all possible pairs with each other. To get a final value, in both cases, the results are simply averaged.

P@k 4.7 is a metric considering the number of top-k items, that are relevant. It is denoted as

$$P@k = \frac{\text{Number of relevant items in } K}{\text{Total number of items in } K} \tag{4.7}$$

and an easy-to-understand metric with high interpretability. Nevertheless, its downside is that the quality of the ranked results is not considered, since the exact position which relevant items occupy is not taken into account.

Another typical IR metric is the **nDCG@k** 4.8, comparing the ranking of the retrieved items with their ideal ranking. It is denoted as

$$nDCG@k = \frac{DCG@k}{IDCG@k}, \tag{4.8}$$

whereas DCG represents the discounted cumulative gain, and its ideal counterpart ($IDCG$), where all relevant elements are at the start of the list. The DCG simply represents the number of relevant items within the top-k items, penalised by a logarithmic discount if the item appears further down in the ranked list. This $DCG@k$ score is then normalised by the ideal ranking. The $nDCG@k$ is a rank-aware metric, handling binary and numerical scores. Nevertheless, the interpretation might be limited since the logarithmic discount could appear arbitrary.

4.4. *TrialLlama*

4.4.1. Fine-tuning

Llama2 was fine-tuned to create *TrialLlama* and compared to a version fine-tuned on a different hyperparameter set and the pre-trained model. For fine-tuning, the training dataset samples had to be transformed into prompting items in a dedicated step. These

prompting items were generated in two steps: 1) Tokenising the inputs and masking the labels, and 2) injecting the tokenised items to a prompt template in the following format

```
"prompt_input": "{instruction} {topic} {clinical_trial} {response}".
```

These prompting items are generated during training, which results in less memory consumption since the items don't have to be created and saved temporarily beforehand.

The validation dataset was used to select the best model. LoRA was used as the underlying PEFT method and all weights were loaded in 8-bit. Training metrics such as *average_train_epoch_loss* and *average_eval_epoch_loss*, etc., were tracked using MLflow.

After approximately 200 experimental runs with different hyperparameter settings and dataset sizes, the most adequate set of hyperparameters was selected for *TrialLlama*. Table 4.5 shows an overview of tested hyperparameters, building up on each other and always selecting the best runs.

instructions	12
dataset sizes	10
gradient accumulation-lr combos	8
weight decays	2
dataset balancing strategies	2
one-shot strategies	2

Table 4.5.: Number of different versions of dataset or hyperparameter-related variables tested while experimentation.

Besides, since most *Llama2*-based prototypes found in literature simply utilised the hyperparameters proposed in the original work of Touvron, Martin, and Stone [4] for finetuning. Hence, besides training *TrialLlama*, a version called *TrialLlama Meta* was trained with the originally proposed hyperparameters.
Further, the conducted experiments also included adding a one-shot example already in the fine-tuning stage, which was omitted since train-valid performance did not increase significantly. The best-performing hyperparameters for training *TrialLlama* and the hyperparams for *TrialLlama Meta* are displayed in Table 4.6. The LoRA configuration was kept the same for both models (Table 4.7).

The best performing *TrialLlama* version was fine-tuned on 3000 samples equivalent to 3.01 million tokens. Experimentation revealed decreased performance when fine-tuning was conducted on more samples. A micro-batch size of four was chosen with no gradient accumulation, resulting in an effective batch size of four as well. *TrialLlama* achieved its optimal validation loss after three epochs of training, starting to overfit afterwards.

4.4.2. Evaluation Details

All introduced performance metrics were calculated for the original three-label classification and a transformed ***binary problem***, achieved by mapping the two negative labels *excluded* and *irrelevant* to the same numerical value. Since two of the three labels are

	TrialLlama	*TrialLlama Meta*
epochs	3	
train samples	2700	
valid samples	300	
train tokens	3.1M	
micro batch size	4	
max tokens	2048	
weight decay	0.1	
optimiser	AdamW	
one-shot example	False	
lr	1e-4	2e-5
gradient accumulation	1	16
learning rate scheduler	stepLR	cosineLR
β_1, β_2	N/A	0.90, 0.95
min lr	N/A	2e-6

Table 4.6.: Hyperparameters utilised for the fine-tuning of *TrialLlama* compared *TrialLlama Meta* fine-tuned on the hyperparameters proposed by Touvron, Martin, and Stone [4]

Hyperparameter	*Value*
r	8
alpha	32
task type	CAUSAL_LM
lora dropout	0.05

Table 4.7.: Hyperparameters for LoRA optimisation.

negative, the binary mapping and the calculation of binary performance metrics of the models' responses can be justified as valid strategy.

TREC trec_eval evaluation

This *trec_eval* script takes the provided TREC gold-label file for a specific conference year and teams top-1000 retrieved patient-trial pairs from their best run as input. This run file underlies the constraints of being in the specified format. Then the script, written in *C*, calculates and returns typical IR metrics.
As mentioned in Chapter 3, two types of runs exist in the TREC Clinical Trials track. The automatic and the manual run. The automatic run tests the model-only classification capabilities, while the manual runs includes a manual intervention of any kind to enhance the retrieval results. This work only considers automatic runs for evaluation, since no HITLS strategies were applied.

4.5. Hardware Setup

The model's fine-tuning was run on a single NVIDIA RTX A6000 48GB GPU. Hence, initial experiments were conducted to find the maximum number of tokens which fit on 48GB with the optimal *batch size* of four. This resulted in a maximum token number of roughly 2300 tokens. To align with the convention of 2^x for the number of tokens, the

maximum number of tokens for fine-tuning was fixed to 2048 tokens, corresponding to approximately 1000 words. This led to a peak GPU utilisation of 40.7 GB.

For the inference the maximum token number of 4096 tokens was applicable, resulting in a peak GPU utilisation of 22.4 GB, since for inference an effective *batch size* of one is the case. It is important to mention, that the maximum number of tokens includes the number of input tokens as well as the number of generated tokens. Hence, the input token number was limited to 3072 (approximately 1500 words) which is enough to fit most of the prompt items from the testing dataset. The residual 1024 tokens were reserved for the generation process. The fine-tuning of *TrialLlama* around 30 hours.

4.6. Results and Discussion

TrialLlama disclosed strengths and weaknesses about the fine-tuning procedure, the extraction and patient-trial matching capabilities of *Llama2* and the TREC Clinical Trials track datasets in general.

4.6.1. Findings

It was possible to fine-tune *Llama2* in a supervised manner provided TREC Clinical Trials track conference data and labels, even though requiring substantial efforts in instruction crafting and inference-related hyperparameter adjustments. Yet, the final version of *TrialLlama* showed decent performance, especially in the binary evaluation setting. With an accuracy of 0.812 *TrialLlama* achieved solid results including or excluding a patient from a given CT compared to its baseline and *TrialLlama Meta*. In the following *TrialLlama* is referenced simply as *TL* for the sake of shortness and simplicity.

Classification Performance Comparison

Table 4.8 shows the performance comparison of *TL*, *TL Origin* and the baseline for the three-label problem and Table 4.9 for the binary problem. The binary results stand out, especially in terms of precision, scoring 0.927. Nevertheless, all other metrics show good results as well with an Accuracy of 0.813, an F1 score of 0.883 and an AUROC of 0.751, indicating good discrimination capabilities. Nevertheless, *TrialLlama* also performs significantly better in the three-label classification setting than the compared models reaching an accuracy of 0.627. The AUROC score stays almost the same with 0.716.

While testing different model versions on inference, different numbers of empty or unusable responses, containing unnecessary characters and symbols, were noticed. This was caused by the fact that none of the fine-tuned models were capable of responding with exactly one of the three provided labels. This led to the problem of non-matched items by the regex, leading to a total of 287 non-evaluable responses. Yet, based on the full testing dataset size of 5,536 items, fine-tuning the model on optimal hyperparameters reduced the number of non-evaluable responses by 67% and 81% compared to *TL Origin* and the baseline, respectively.

Additionally, comparing the inference results of *TrialLlama* without a one-shot example reduced the number of *evaluable items* drastically by roughly 30% to 3,556. The utilised one-shot examples for the classification and the reasoning task are attached to Appendix A.

	Accuracy	Precision	Recall	F1	AUROC	Evaluable Items
Baseline	0.162	0.054	0.333	0.093	0.500	3890
TL Meta	0.247	0.351	0.346	0.214	0.506	4689
TL	**0.634**	**0.532**	**0.595**	**0.530**	**0.716**	**5234**

Table 4.8.: Three-label problem classification performance of *TrialLlama* (TL) vs. Baseline on all available testing samples.

	Accuracy	Precision	Recall	F1	AUROC	Evaluable Items
Baseline	0.162	0	0	0	0.5	3890
TL Meta	0.312	**0.847**	0.223	0.353	0.504	4689
TL	**0.813**	**0.926**	0.843	**0.883**	**0.751**	**5234**

Table 4.9.: Binary problem classification performance of *TrialLlama* (TL) vs. Baseline on all available testing samples.

While training *TrialLlama* achieved a minimal *training loss* of 0.125 and a *validation loss* of 0.175 before starting to overfit as seen in the train vs. validation plot in Figure 4.6.

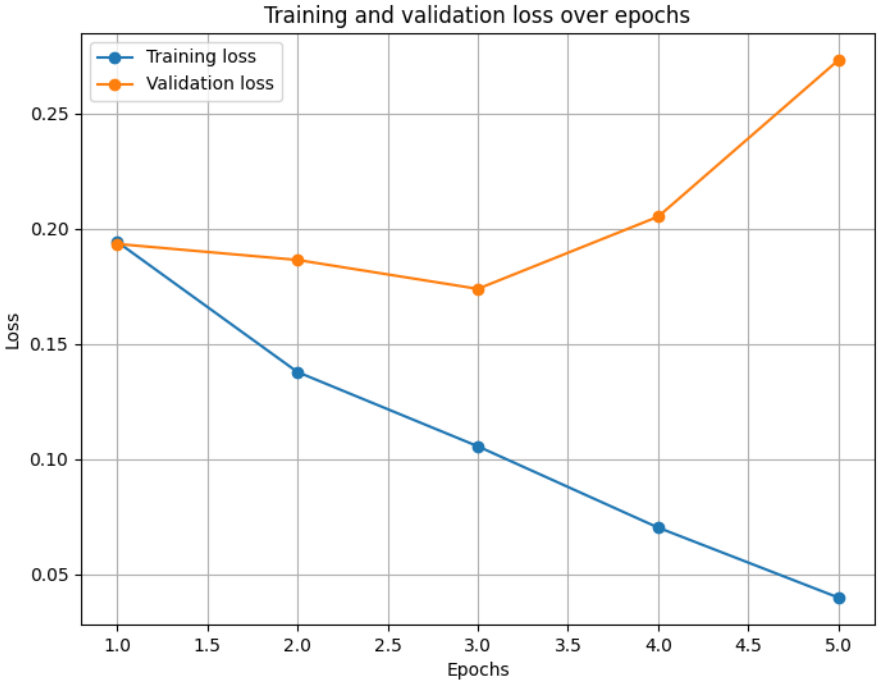

Figure 4.6.: Training vs. validation loss of *TrialLlama* training.

Taking a look at the confusion matrix for the three-label classification in Figure 4.7 the model performed decently on the correct prediction of the *irrelevant* class with 64.65% of correctly classified *irrelevant* patient-trial pairs. TL struggles most in differentiating

between *irrelevant* and *excluded*. A share of 23.43% actually *irrelevant* items were classified as *excluded*. Only 11.91% of *irrelevant* pairs were classified as eligible. Given the strong predominance of the *irrelevant* class in the testing dataset, the model performed fairly decent. For the *excluded* class the model struggles the most, predicting equally between *excluded* and *eligible*, only getting 44.52% correctly. *TL* performs decently on the *eligible* classification of patient-trial pairs, with a correct classification of 66.82%, scoring highest among the three classes.

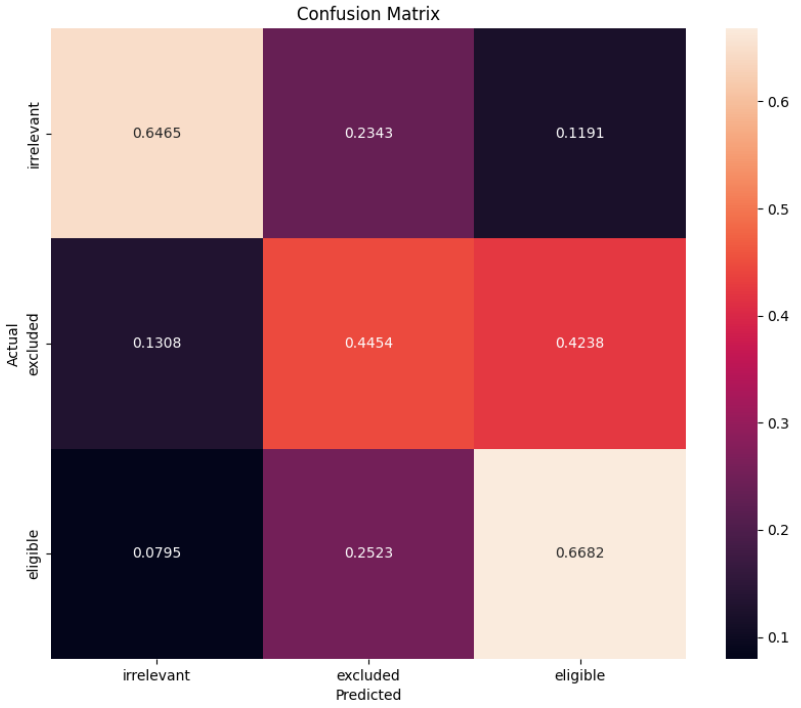

Figure 4.7.: *TrialLlama* confusion matrix for the three-label classification inference.

Contemplating the binary confusion matrix in Figure 4.8, where both negative classes were merged, a significantly better result can be observed. The binary *excluded* class (*irrelevant* plus *excluded*), achieves a correct classification of 84.06%. The *eligible* result did not change, since only the negative labels were merged. Yet, the model's struggles with *eligible* patient-trial pair classification can be seen more clearly now since the binary *confusion matrix* highlights the 33.18% of actually eligible pairs which were wrongly classified as *excluded*.

4.6.2. Comparison to Similar Systems

As for the TREC teams, *TL* was run on the datasets for 2021 and 2022 individually. Table 4.10 shows the distribution of the testing dataset classes split by year. To compare *TL* with these teams, the summed token probabilities of the classification were utilised for a relevancy ranking of the classified items.

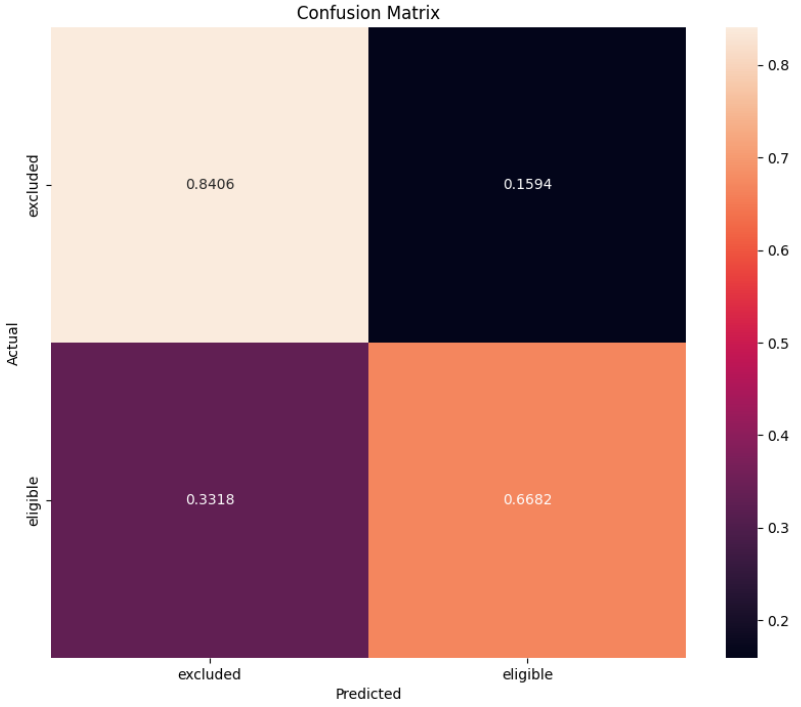

Figure 4.8.: *TrialLlama* confusion matrix for the binary classification inference.

	2021	**2022**
eligible	747	184
excluded	549	74
irrelevant	2139	1845
Total	3435	2203

Table 4.10.: Samples for inference from the testing dataset divided by year.

A strong imbalance can be seen between the class *irrelevant* and the other classes, especially for 2022. In general, this large imbalance is caused by the fact, that most CTs are irrelevant for a patient. In this work's case, the heavy imbalance in 2022 was amplified by the random selection of patient descriptions and their respective CTs for while splitting the training and testing datasets. Unfortunately, the randomly selected patient descriptions from 2022 are mapped mostly to irrelevant CTs.

Table 4.11 and Table 4.12 compare the results of TL and the best-performing teams of the TREC for 2021 and 2022, respectively. The results show that *TL* achieved the best nDCG@10 and P@10 scores compared to the best-performing TREC Clinical Trials track teams from the past two years. Scores were derived from the overview publications of the TREC Clinical Trials track 2021 and 2022 [9].

One can see decent results comparing *TL* 2021 with the winners of the TREC Clinical Trials track from 2021, where *TrialLlama's* nDCG@10 score of 0.652 is relatively close to

	nDCG@10	P@10
h2oloo 2021	**0.712**	0.593
TL 2021	0.652	**0.738**

Table 4.11.: Comparision of best TREC Clinical Trials track runs with *TrialLlama* (TL) on topics from 2021.

	nDCG@10	P@10
h2oloo 2022	**0.613**	**0.509**
TL 2022	0.328	0.375

Table 4.12.: Comparision of best TREC Clinical Trials track runs with *TrialLlama* (TL) on topics from 2022.

the nDCG@10 score of *h2oloo 2021* team with 0.721 and even surpasses the 2021 team's P@k value with 0.738. This shows the decent retrieval capabilities of *TrialLlama* not considering the ranking quality of the results.

Regarding *TL* 2022, the result is quite poor, achieving a nDCG@10 of 0.328 compared to the *h2oloo* teams value of 0.613. Also the P@10 scores significantly below the *h2oloo* team. Overall, the non-LLM-based solutions of the TREC Clinical Trials track teams perform better compared to *TL*.
The *trec_eval* script output for both years is attached to Appendix B.

A comparison between *TL* and its closest counterpart *TrialGPT* was not possible to the full extent, since *TrialGPT* was evaluated on 185 topics and 18,238 CTs in a criterion-level manner from three different datasets, including the TREC Clinical Trials track datasets and SIGIR Conference Challenge dataset[6]. The SIGIR dataset was not used for this work's inference, since the patient descriptions are provided in slightly different formats requiring adjustments for data pre-processing. Yet, restricted comparison can be justified, since SIGIR topics were extracted from the TREC Clinical Decision Support track (2014/2015), being sufficiently similar to the topics provided in TREC 2021/2022. Hence, comparison under this assumption can be justified.

Nevertheless, a comparison with *TL* was conducted on as much testing data as was available in this work. The results can be seen in Table 4.13. *TrialGPT* performs significantly better in terms of nDCG@10 and P@10. Yet, *TL's* performance can be argued as decent, considering the more generic trial-level evaluation approach and the much smaller model size (13B vs 185B).

	nDCG@10	P@10
TrialGPT	**0.748**	**0.675**
TL	0.551	0.571

Table 4.13.: Comparison of *TrialLlama* (TL) and *TrialGPT*.

Overall, *TL* performed solidly among TREC Clinical Trials track-based solutions and its

[6]https://data.csiro.au/collection/csiro:17152, accessed 26.01.24, 16:48

closest counterpart *TrialGPT*. Nevertheless, the comparison was limited due to the unique approach of *TrialLlama*, requiring further evaluation in future work.

4.6.3. Limitations

Architecture of pre-trained Model

Since *TL* is based on *Llama2*, which is a decoder-only Causal Language Model, capable of generating text in predicting one token at a time given an input sequence. This does not state optimal conditions for the classification task faced in RQ1, where Masked Language Models like BERT [15] or *Seq2Seq* models like BART [65] might be the better choice as used in most of the approaches of TREC Clinical Trials track 2021 and 2022 and non-TREC related publications. A possible solution for this issue could be an approach to adapt LLMs for fine-tuning with discriminant labels to perform supervised label prediction by Li, Li, Liu, *et al.* [66]. Yet, this was not explored or adapted in this work.

Yet, forcing the model to answer with a label only revealed itself to be more tricky than first expected, requiring excessive hyperparameter and instruction experimentation. After fine-tuning, adding a one-shot example to the testing items became mandatory to trigger the model's chat capabilities. Simply changing the *max new tokens* parameter or the instruction did not dissuade the model from responding in the classification.

Comparability

Due to computational limitations and the supervised fine-tuning approach only a substantially smaller number of dataset items was available for classification, compared to the TREC Clinical Trials track team runs. Usually, the TREC teams fine-tuned their models on different data sources or in an unsupervised manner, since gold labels were created based on the teams' submissions and only provided after the conference took place. Therefore, all patient-trial combinations are available for final inference and the resulting relevancy ranking of the top 1000 items. In this work's case, most data was already utilised for model fine-tuning or removed during pre-processing due to length or unsuitable structure, and hence not used for inference. This could be one reason for the poor results for the 2022 comparison since many of the relevant patient-trial pairs possibly weren't included in the testing dataset due to the arbitrary training-testing split. Hence, the comparison results should be viewed carefully having limited expressiveness.

Even though *TrialGPT* is the closest relative to *TrialLlama*, the comparison is possible only in a limited way since the approaches for evaluation were very different. *TrialGPT* focused on a very thorough, mostly manually conducted qualitative evaluation on a criterion level, whereas this work aimed primarily at an automatic evaluation strategy of *TrialLlama*, stating the classification task on trial-level. Also, *TrialGPT* was fine-tuned with an *in-context* learning approach.

Lastly, based on the unique architecture of *TrialLlama*, a comparison to COMPOSE by Gao, Xiao, Glass, *et al.* [54] was not feasible due to completely different datasets utilised for the final classification inference, which were not publicly available.

Hyperparameters

The search for optimal hyperparameters turned out to be quite hard, which might be explained by the non-optimal data-architecture fit. Hence, it cannot be guaranteed that the hyperparameters used for this work are optimal, especially since the validation loss reached an unsatisfying minimum before overfitting and increasing again. Nevertheless, the chosen hyperparameters showed promising fine-tuning results, which one can build upon.

Dataset

Also, there are shortcomings in data quality, which impacted the fine-tuning issue of the model. While working with the dataset, a strong resemblance among some topics and the high discrepancy between the number of available topics and the total number of judged CTs was noticed. With the already relatively small number of unique patient descriptions, the high repetition of topics within the training samples resulted from this. As Jin, Wang, Floudas, *et al.* [59] already concluded the provided TREC datasets might be over-simplified and hence not very representative of real-life settings.

5. Patient-Trial Reasoning

The following chapter builds upon the preceding chapter utilising the fine-tuned *TL* model for the reasoning task, introduced in Section 5.1, explaining the major differences for the reasoning task. Section 5.2 provides details about the evaluation strategy. Lastly, in Section 5.3 the findings are summarised, limitations discussed and some theoretical & practical implications derived.

5.1. Reasoning task

The reasoning task was tested on the best-performing *TL* version fine-tuned for the classification task, to keep the overall model consistent and since there was no dataset available, containing the required human-like responses for a fine-tuning suitable the reasoning task. So for the reasoning task, only some minor changes were made: 1) A dedicated instruction to trigger the more step-by-step extraction and answering capabilities for ECs and 2) some hyperparameter tweaks to re-trigger the chat capabilities of TL, since otherwise, the model would continue to respond in a classification manner.

Instruction and Response Initiator

For the reasoning task, the last sentence of the instruction was modified to "*Please give a final summary at the end*" and the magic words "*Let's think step by step*" were added to the response initiator. This helped to force the model to extract criteria one by one concisely. This slightly different instruction and response initiator was only used for the reasoning task.

Dataset

A small sample of 15 randomly selected items was extracted from the testing dataset. These dataset items, already adjusted with the inference instruction alongside the one-shot example for the reasoning task, are ready for prompting TL and evaluate the responses qualitatively. Additionally, the binary balancing strategy was re-utilised to achieve the best possible class distribution of the extracted dataset samples, considering the relatively small evaluation size.

Hyperparameters

The *max new tokens* value was changed from 10 to 800 to allow for a more detailed response, providing enough room for generation and parallelly contributing to the reactivation of the chat capabilities.

5.2. Evaluation Strategy

5.2.1. Evaluation Details

To evaluate the results of the reasoning task, the pre-trained, as well as TL were prompted individually with the 15 randomly selected patient-trial pairs, resulting in 30 responses.

The task was to extract all criteria from the CT one by one and discuss the patient's eligibility in a step-by-step manner before giving a final verdict.

Firstly, the retrieved responses were rated by two annotators from the IAM (the author included) using a five-point Likert scale ranging from *very poor* (1) to *excellent* (5). For guidance, an annotation guideline (Table 5.1) was developed and provided alongside a short task instruction 5.1.

Expression	Criteria
(1) Very Poor	Wrong answer + no criterion OR Criterion(s) extracted but labeled wrongly Gibberish answers
(2) Poor	Only correct label **OR** wrong label with partly correct criterion/explanation
(3) Fair	Correct label + one criterion extracted/short explanation given Wrong label + multiple criteria correctly extracted and classified/correct reasoning
(4) Good	Correct label + > 2 criterions labeled correctly/relevant explanation given **OR** correct label + single most relevant criterion given
(5) Excellent	Correct label and concise explanation **OR** all relevant criterions extracted and labeled correctly

Table 5.1.: Annotation guideline developed for the raters.

Following Task:

1. Read the patient description,
2. Read the Clinical Trial (CT) Description
3. Read the models response on the eligibility of the patient given the patient-CT combination
4. Rate the response compared to the derived ground-truth reading the patient description and the CT based on the 5-point Likert-Scale
5. Take notes on interesting observations (e.g. model performed poor, was the CT written too complicated?, etc.)

Figure 5.1.: Annotation task instruction containing five steps.

Cohen's kappa κ was calculated for the Inter Annotator Agreement (IAA) [67] with

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{5.1}$$

where p_0 denotes the relative observed agreement between two annotators and p_e describes the hypothetical probability of agreement by chance. This IAA metric was selected since

Cohen's kappa allows the calculation of the inter-rater reliability of exactly two raters, which is the case for this work. Additionally, the average annotator ratings for pre-trained and fine-tuned models were calculated.

Secondly, to get an additional, more feasible qualitative assessment of the models' responses, some performance metrics were calculated manually. This was achieved via an in-depth CT and response evaluation, allowing the extraction of the total number of all ECs contained in 15 samples and the comparison of aspects like number of extracted ECs, the distribution between inclusion and exclusion criteria, correctly classified extracted ECs and the number of correct verdicts. With this information the calculation of the *accuracy*, the *precision*, and the *recall* was possible.

5.3. Results and Discussion

5.3.1. Findings

Solving the reasoning task revealed insights into the great capabilities of small open-source LLMs in extracting and assessing information from input data sources. *TL* showed good performance on the extraction of most of the relevant criteria, pointing out missing or incomplete information and making assumptions based on available information such as deriving a patient's physical state. The experiments also showed the already great capabilities of the pre-trained model in extracting most of the inclusion EC correctly. Yet the pre-trained model stopped there most of the time, neither answering the patient's eligibility for the individual criteria nor giving a final verdict.

Results

Even though TL was fine-tuned on a classification task only, the model showed enhanced capabilities in the correct extraction and classification of ECs and its understanding and assessment of the underlying information.

The outcomes of the qualitative evaluation are summarised in Table 5.2. Since the baseline model sometimes only extracted criteria instead of answering the question of eligibility, those "empty" answers were counted and compared as well.

The 15 testing samples contained 10 *eligible*, two *irrelevant* and three *excluded* patient-trial pairs, with a total of 160 ECs.
The number of not-answered patient-trial classifications dropped from eight to zero. *TL* showed enhanced performance in extracting criteria, compared to the baseline. *TL* extracted 74 criteria, as opposed to 56 in the baseline.

Further, the accuracy of correctly classified patient-trial pairs increased by 0.666, The precision increased by 0.640 and the recall by 0.319 compared to the baseline. In the number of total extracted ECs, TL only enhanced little to the baseline by 0.125. Looking at the class distribution of extracted ECS, a bias towards extracting only inclusion criteria

can be noticed in the baseline (43 vs. 20), which is not resolved after fine-tuning (47 vs. 27).

Lastly, for the two annotators (A1, A2) a Cohen's Kappa of 0.353 was calculated, indicating a fair agreement. It is noticeable that the *TL* enhanced regarding the average rating (1-5) by both annotators, exceeding the average rating score for the pre-trained model by 2,2 points. Nevertheless, among the 30 evaluated responses, only one single item had a deviation of ±2 between the annotators' ratings. 14 items matched perfectly between annotators and 16 varied by only ±1 point (Table 5.3).

	Llama2	*TrialLlama*
Total number of responses	15	
No. of responses with eligibility verdict	7	15
Total ECs	160	
Retrieved ECs	63	74
Inclusion Criteria	43	47
Exclusion Criteria	20	27
Accuracy of verdicts	0.257	0.933
Precision	0.333	0.973
Recall	0.131	0.450
Cohen's Kappa	0.353	
Average annotator score	2.667	4.467

Table 5.2.: Qualitative evaluation of the base model vs. *TrialLlama* on 15 randomly selected samples from the testing dataset, adjusted by modifying the instruction and the response initiator.

A1/A2	Excellent	Good	Fair	Poor	Very Poor	Row Sum
Excellent	6	4	0	0	0	10
Good	4	1	1	1	0	7
Fair	0	0	2	1	0	3
Poor	0	0	0	4	4	8
Very Poor	0	0	0	0	2	2
Col Sum	10	5	4	6	6	30

Table 5.3.: Summary of rating results of both annotators utilised for calculation of Cohen's Kappa.

Inference Response Examples

Figure 5.2 depicts an exemplary model response and the respective patient-trial pair. The ground truth is *eligible*.
One can notice that the model, even though answering as enumeration, did not insert any kind of separation between inclusion and exclusion criteria, making it hard for any human reader to understand which criteria belong to which group, requiring the original CT ECs as reference.

Nevertheless, very interesting to notice are the model's great extraction and reasoning capabilities. First of all, *TL* correctly derived the patient to have a *hypothyroidism* based

on her symptoms and laboratory results (highlighted in yellow). Also *TL* was capable of discriminating between past and current values as seen in the example (highlighted in green) since the patient description mentions the BMI before and after Levothyroxine administration over a certain period of time, which helped the patient to lose weight. However, she's still eligible based on her new BMI, as correctly concluded by *TL*. Further, the model reasons logically about the patient being "sufficiently substituted" since she started to take Levothyroxine to treat her hypothyroidism (purple).
Yet, two things get a bit confusing here, since the CT says either a) not sufficiently substituted patients, or b) patients sufficiently substituted with a TSH in the range of 0.5 to 2.5 mlU/D. Hence, it is unclear if *TL* considered this information as well or simply landed a lucky shot this time.

Lastly, for the exclusion criteria, TL summarised the exclusion criteria by simply stating that the patient does not meet any of them (orange). This reflects the issue noticed in the findings of not extracting exclusion criteria equally detailed as inclusion criteria. Nevertheless, in this example's case, the model's answer is not wrong since it is true that the patient at hand does not meet any of the stated exclusion criteria. At least, they are not mentioned in the patient's description Yet, summarising the evaluated responses, TL shows astonishing results regarding criteria extraction, reasoning and final verdict statement.

The output files containing all evaluated model responses, patient description and CTs can be found as part of the Github repository[1]. More response examples, with their corresponding CT and are attached to Appendix C.

5.3.2. Limitations

Evaluation Scope

The evaluation of the reasoning task was only conducted on a small sample by non-medical professionals and compared to the pre-trained model's responses. Hence, the results do not reflect the reasoning capabilities of the fine-tuned model against bigger models such as TrialGPT, *ChatGPT* or even *GPT4*. Also, no tests in the clinical environment have been conducted yet, due to a lack of data accessibility, doctors' availability and the currently very slow inference of the model taking up to five minutes depending on the patient-trial pair.

Hallucination

The widely known issue of hallucination occurred in some cases. This problem can be attributed to the relatively small number of patient descriptions and some resemblance between descriptions, which led to a somewhat repetitive dataset composition. The hallucination happened in some instances, resulting in the model's responses reflecting knowledge from training data, only considering the actual information from the underlying CT partly or not at all.

[1] https://github.com/UKEIAM/TrialLlama/tree/main/qual$_e$*val*$_o$*utput*

Response

1. Does the patient have primary hypothyroidism? Yes, the patient has primary hypothyroidism, as evidenced by her symptoms of fatigue, weight gain, and abnormal spotting between menses, as well as her laboratory results showing high TSH and low free T4 levels.
2. Is the patient's BMI within the study's inclusion criteria of 19 to 35 kg/m2? Yes, the patient's BMI of 31 is within the study's inclusion criteria.
3. Does the patient have a documented diagnosis of primary hypothyroidism? Yes, the patient has a documented diagnosis of primary hypothyroidism based on her symptoms and laboratory results.
4. Is the patient currently not sufficiently substituted? No, the patient is currently being treated with Levothyroxine, which suggests that she is sufficiently substituted.
5. Does the patient meet any of the exclusion criteria? No, the patient does not meet any of the exclusion criteria. Therefore, the patient is eligible for the clinical

Patient-Trial Pair

The patient is a 33-year-old woman complained of fatigue, weight gain and abnormal spotting between menses. No hirsutism or nipple discharge was detected. Her BMI was 34. Her lab results were remarkable for high TSH level (13 mU/L) and low free T4 level (0.2 ng/dl). Her anti-TPO levels were extremely high (120 IU/ml). She was diagnosed with Hashimoto's thyroiditis. Her aunt, brother and mother have the same disease. After starting 250 mcg Levothyroxine per day, her symptoms improved significantly and her periods are normal. She is still overweight with BMI of 31. Her most recent thyroid profile revealed all results except for anti-TPO within the normal range: TSH: 2.35 mU/L Free T4: 2.7 ng/dl Anti-TPO: 75 IU/ml

Title: Metabolomics of Thyroid Hormones
Summary: The aim of this study is to develop a metabolome signature of thyroid hormone status. The metabolome signature could be useful in diagnosis and treatment of thyroid dysfunction diseases, especially in cases where TSH cannot be reliably used.
Intervention Type: Diagnostic Test
INCLUSION CRITERIA: All participants: - Age 18 to 80 years - BMI 19 to 35 kg/m2 - Informed consent as documented by signature A) Patients with primary hyperthyroidism TSH < 0.2 miU/ and free thyroxine (fT4) > 25 pM or fT3 > 8 pM B) Patients with primary hypothyroidism, currently not sufficiently substituted TSH > 8 milli-International unit (mIU)/1 or fT4 < 10 pM C) Patients with diagnosis of primary hypothyroidism, sufficiently substituted Documented diagnosis of primary hypothyroidism Substitution with L-Thyroxin TSH in target range between 0.5 and 2.5 mlU/ D) Patients with secondary hypothyroidism Pituitary disease with documented secondary hypothyroidism
EXCLUSION CRITERIA: - Diabetes mellitus (HbA1c >6.5%) - Severe concomitant diseases: chronic heart failure, liver cirrhosis, kidney failure, active cancer - Abuse of alcohol or illicit drugs - Women who are pregnant or breast feeding - Inability to follow the procedures of the study, e.g. due to language problems, psychological disorders, dementia, etc. of the participant - Previous enrolment into the current

Figure 5.2.: *TrialLlama* example reasoning task response.

Inclusion Criteria Bias

Lastly, as already noticed, the exclusion criteria are not focused on as strongly as the inclusion criteria by the model, resulting in a very short exclusion criteria statement. This behaviour was noticed in most evaluated responses, with only two cases, where the exclusion criteria were properly extracted. Nevertheless, since the pre-trained and fine-tuned version clearly shows good capability of extracting required information, this inclusion criteria bias hints at issues with the one-shot example, the instruction or some other issues with the CTs.

5.3.3. Theoretical & Practical Implications

Theoretical aspects

From a theoretical perspective the data used for training *TL* can be argued as adequately domain-specific and with the evaluation of the reasoning task showing decent performance and comprehensible responses in the model's reasoning skills, some issues mentioned by Thirunavukarasu, Ting, Elangovan, *et al.* [28] (*Accuracy*, *Coherence*, *Security and Privacy*) were successfully tackled with this work. This opens the door for further research towards a patient-trial matching assistant, which could be used by physicians in their daily work routine, reducing time spent on the task of patient-trial matching.

Further, the decent performance of TL on the classification task showed that small-sized LLMs can be forced to act as classifiers. In this case for patient-trial matching.
But more promising are the results of the reasoning task. Even though *TL* was solely fine-tuned on a classification task, *TL* demonstrated great enhancements in its extraction, discussion and deduction capabilities compared to the baseline. Since this was achieved by only a few inference-related adjustments such as increasing the *max new tokens* parameter, enriching the instruction and adjusting the response of the one-shot example, the need for a dataset annotated by experts is rendered not absolutely necessary. This is a positive finding since these expert datasets are rarely available and effortful in creation. Nevertheless, a dedicated expert dataset would most probably enhance the model's performance significantly.

Practical aspects

TL opens the doors for deployment within real clinical context since stating no privacy concerns allowing for being trained and deployed locally. Its small size even allows it to run inference on consumer-grade hardware, requiring as little as 22GB of RAM. Further, the deployment in the clinical environment holds great benefits for model fine-tuning since huge amounts of real clinical data, unique for the institution become available. This allows for tailored model development and system integration. One could even think further and consider approaches like federated or split learning, where only the local model weights are shared by multiple clinics, keeping the data private and allowing for the training of a powerful assistant, which could be used beyond patient-trial matching. Additionally, given *TL's* flexible architecture, the provided codebase enables an easy exchange of utilised base models, allowing for experimentation with other pre-trained LLMs, if required.

Keeping the small model size of *TL* has even more advantages like lower energy consumption and cheaper hardware, which is not irrelevant since the former is going to play a more and more important role, considering green energy goals and the trend towards more sustainable ML in general.

Rapid evolution of LLMs in medical applications

Nevertheless, one should keep in mind that the domain of LLMs for medical applications is changing rapidly. With the release of systems like *GPTs* by OpenAI [2], *Gemini* by Google[3] and new fine-tuned models like *Meditron* [39], new promises are made and new possibilities are created. Since OpenAI promised local, privacy-preserving deployment of a specialised GPT for enterprise customers, the possibility of using such a system in the clinical context has to be evaluated as well as for a deployment of the *Meditron* model.

With the given insights while developing both *TL*, this work pushes research further towards more capable patient-trial matching systems, preserving the chat capabilities of LLMs and previewing future assistance systems for clinicians.
In conclusion, this work"s proposed open-source prototype is practically utilisable for the patient-trial matching task. Additionally, the provided open-source repository for the preprocessing of the TREC Clinical Trials track data, fine-tuning a pre-trained LLM, inferencing and evaluation, can help in building further real-world LLM applications, especially for the domain of clinical assistance systems.

[2] https://openai.com/blog/introducing-gpts, accessed 13.11.23 21:37
[3] https://blog.google/technology/ai/google-gemini-ai/capabilities, accessed 10.01.24 15:56

6. Future Work

This chapter gives an outlook on future work, addressing issues and ideas which came up during this work's development. Section 6.1 discusses more ideas on *TrialLlama's* architecture, continued by a discussion about the dataset composition in Section 6.2. In Section 6.3 this chapter is closed with some thoughts on more elaborate hyperparameter tuning.

6.1. Architecture

The proposed architecture does not contain any feedback loops or correction systems. A common approach for the fine-tuned model's alignment is RLHF. Since this approach is very effortful [16], one could think about exploring new approaches like Reinforcement Learning from AI Feedback. Additionally, the systems architecture could profit from access to a broader knowledge base. This could be achieved by exploring RAG approaches, allowing the model to fetch recent data from a defined database, enhancing its understanding of concepts and terms and delivering the most recent results in the fast-changing world of medical applications terminology.
Further, the Causal Language Model architecture of this work's base model is not ideal for the classification task. Nevertheless, it is very suitable for the reasoning one. Hence, experimenting with the approach of Li, Li, Liu, *et al.* [66] could be of high interest, to see how the model performs after adjusting it to label-supervised fine-tuning.

Lastly, other small-sized open-source LLMs should be implemented with this work's architecture. Running the evaluation on these different models would allow for a bigger comparison and lead possibly to finding the best base model for the real clinical deployment of a patient-trial matcher or similar applications.

6.2. Dataset Composition

One important point is to rethink the data generation pipeline to achieve an even cleaner and more curated training dataset, leveraging the full power of the current architecture and allowing for more precise learning of relevant data. Due to the sometimes quite messy and complicatedly written CT descriptions paired with patient descriptions and the instructions, the created training and testing datasets used for this work imposed different challenges in the LLM fine-tuning due to input length, unknown words, laboratory values and mathematical expressions. Hence, it should be analysed what parts are attended most by the model, since the inputs could be reduced to more relevant parts of the text, enhancing the system's overall speed and performance.

6.3. Hyperparameter Tuning

Due to time and computation limitations, hyperparameter tuning was only conducted to a reasonable extent. This said, there might be better hyperparameter combinations to

increase this work's prototype's classification capabilities. The fact, that most related work with similar LLM-based approaches did not search for optimal hyperparameters adopting the hyperparameters described in the original *Llama2* paper, highlights the need for more hyperparameter tuning. Especially in the case of utilising PEFT optimisation methods, these hyperparameters might not be optimal. Concluding, more hyperparameter optimisation has to be conducted. Even if Han, Adams, Papaioannou, *et al.* [42] pointed out a strong performance decrease in utilising PEFT methods, only a few researcher groups have access to resources for fine-tuning a model in full size. This underlines the need for better optimisation strategies and the increasing demand for such in LLM research.

7. Conclusion

7.1. Research Questions Revisited

> *RQ 1: Can a small-sized open-source LLM be fine-tuned to achieve feasible results in classifying eligibility among similar patient-trial matching models?*

Based on the results of the classification task, RQ1 can be answered with yes. A small-sized LLM such as *Llama2* is capable of classifying a patient's eligibility given a patient-trial pair with comparable results to non-LLM-based and LLM-based solutions. The best performance is achieved when transforming the original three-class problem into a binary one. This binary contemplation can be argued as sufficient since the outcome is the same: Either a given patient is being included in a CT or not. Additionally, the TREC Clinical Trials track follows a similar approach, where only nDCG@k is calculated in the context of a three-class problem and all other metrics in the binary approach. Hence, the three-label problem can be argued as unnecessary and can be reduced to a binary one, especially for the use case of patient-trial matching in the real clinical context.

> *RQ 2: Can a small-sized fine-tuned LLM give reasonable rationals on its eligibility decision for patient-trial matching?*

Even though *TL* was fine-tuned on a classification task given the available datasets, it performed quite well solving the reasoning task outpacing the baseline easily. Even though the sample size for the reasoning task answering RQ2 was relatively small, the results are very promising. The already good capabilities of the baseline in extracting relevant information from quite complex inputs, utilising an adjusted instruction and one-shot example were enhanced significantly by fine-tuning. Nevertheless, future work is needed.

7.2. Final Words

This work explored the realms of automated patient-trial matching fine-tuning *Llama2* and proposing the open-source available *TrialLlama* and its codebase.
TrialLlama was tested on two different tasks: 1) the patient-trial eligibility classification task and 2) the reasoning task, extracting and classifying ECs one by one discussing the criteria and giving a final verdict on a patient's eligibility.

To the best of my knowledge, this work is the first of its kind which explored the end-to-end patient-trial matching capabilities of small-sized open-source LLM, providing first insights for model architectures which can be fine-tuned and deployed in real-world clinical environments, addressing privacy concerns and computational limitations, as well as considering energy consumption.

The code for *TrialLlama* is available open-source, as are the weights of the *TrialLlama* [1] and the utilised datasets for training and testing[2].

[1] https://huggingface.co/Kevinkrs/TrialLlama
[2] https://huggingface.co/datasets/Kevinkrs/TrialLlama-datasets/tree/main

By providing these resources, this work hopes to enable other researchers and projects to fine-tune specialised models swiftly, allowing for the fast development of useful solutions leveraging small-sized LLMs. Additionally, the provided repository aims to convey a feeling for the process of LLM fine-tuning, allowing interested people to get started quickly and easily. This work will hopefully be the driver for a faster transition from research to practice and, at best, increased patient care while decreasing the strain on physicians simultaneously, paving the path towards valuable human-AI interaction in the clinical world.

It is important to highlight, that this work does not propose AI systems as replacements for human physicians, but on the contrary campaigns for the side-by-side interaction between humans and machines as the driver for faster and better innovation and optimal patient care.

Bibliography

[1] K. Lauk, M.-C. Peters, J.-L. Velthaus, S. Nürnberg, and F. Ueckert, "Use of Process Modelling for Optimization of Molecular Tumor Boards," en, *Applied Sciences*, vol. 12, no. 7, p. 3485, Jan. 2022, ISSN: 2076-3417. DOI: 10.3390/app12073485. [Online]. Available: https://www.mdpi.com/2076-3417/12/7/3485 (visited on 04/12/2023).

[2] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. P. Campbell, "Introduction to Machine Learning, Neural Networks, and Deep Learning," en, *Neural Networks*, Feb. 2020.

[3] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention Is All You Need*, en, Aug. 2017. [Online]. Available: http://arxiv.org/abs/1706.03762 (visited on 10/24/2023).

[4] H. Touvron, L. Martin, and K. Stone, "Llama 2: Open Foundation and Fine-Tuned Chat Models," en, 2023.

[5] E. J. Hu, Y. Shen, P. Wallis, *et al.*, *LoRA: Low-Rank Adaptation of Large Language Models*, Oct. 2021. DOI: 10.48550/arXiv.2106.09685. [Online]. Available: http://arxiv.org/abs/2106.09685 (visited on 07/25/2023).

[6] K. Zhang and D. Demner-Fushman, "Automated classification of eligibility criteria in clinical trials to facilitate patient-trial matching for specific patient populations," en, *Journal of the American Medical Informatics Association*, vol. 24, no. 4, pp. 781–787, Jul. 2017, ISSN: 1067-5027, 1527-974X. DOI: 10.1093/jamia/ocw176. [Online]. Available: https://academic.oup.com/jamia/article/24/4/781/3038205 (visited on 02/13/2024).

[7] M. R. Kosorok and E. B. Laber, "Precision Medicine," *Annual Review of Statistics and Its Application*, vol. 6, no. 1, pp. 263–286, 2019. DOI: 10.1146/annurev-statistics-030718-105251. [Online]. Available: https://doi.org/10.1146/annurev-statistics-030718-105251 (visited on 11/18/2023).

[8] C. Luchini, R. T. Lawlor, M. Milella, and A. Scarpa, "Molecular Tumor Boards in Clinical Practice," English, *Trends in Cancer*, vol. 6, no. 9, pp. 738–744, Sep. 2020, ISSN: 2405-8033. DOI: 10.1016/j.trecan.2020.05.008. [Online]. Available: https://www.cell.com/trends/cancer/abstract/S2405-8033(20)30163-1 (visited on 01/25/2024).

[9] K. Roberts, D. Demner-Fushman, E. M. Voorhees, S. Bedrick, and W. R. Hersh, "Overview of the TREC 2022 Clinical Trials Track," en, 2022.

[10] D. E. Rumelhart and J. L. Mcclelland, "Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1. Foundations," English, Jan. 1986. [Online]. Available: https://www.osti.gov/biblio/5838709 (visited on 01/15/2024).

[11] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, "A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science," en, in *Supervised and Unsupervised Learning for Data Science*, ser. Unsupervised and Semi-Supervised Learning, M. W. Berry, A. Mohamed, and B. W. Yap, Eds., Cham: Springer International Publishing, 2020, pp. 3–21, ISBN: 978-3-030-22475-2. DOI: 10.1007/978-3-030-22475-2_1. [Online]. Available: https://doi.org/10.1007/978-3-030-22475-2_1 (visited on 01/15/2024).

[12] S. Geman, E. Bienenstock, and R. Doursat, "Neural Networks and the Bias/Variance Dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, Jan. 1992, ISSN: 0899-7667. DOI: 10.1162/neco.1992.4.1.1. [Online]. Available: https://doi.org/10.1162/neco.1992.4.1.1 (visited on 01/15/2024).

[13] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4L: Self-Supervised Semi-Supervised Learning," en, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 1476–1485, ISBN: 978-1-72814-803-8. DOI: 10.1109/ICCV.2019.00156. [Online]. Available: https://ieeexplore.ieee.org/document/9010283/ (visited on 01/18/2024).

[14] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*, en. MIT Press, Aug. 2023, ISBN: 978-0-262-04843-9.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv:1810.04805 [cs], May 2019. DOI: 10.48550/arXiv.1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805 (visited on 01/25/2024).

[16] H. Naveed, A. U. Khan, S. Qiu, *et al.*, *A Comprehensive Overview of Large Language Models*, Nov. 2023. [Online]. Available: http://arxiv.org/abs/2307.06435 (visited on 11/05/2023).

[17] X. L. Li and P. Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353. [Online]. Available: https://aclanthology.org/2021.acl-long.353 (visited on 11/24/2023).

[18] B. Lester, R. Al-Rfou, and N. Constant, *The Power of Scale for Parameter-Efficient Prompt Tuning*, Sep. 2021. [Online]. Available: http://arxiv.org/abs/2104.08691 (visited on 11/24/2023).

[19] N. Houlsby, A. Giurgiu, S. Jastrzebski, *et al.*, *Parameter-Efficient Transfer Learning for NLP*, Jun. 2019. [Online]. Available: http://arxiv.org/abs/1902.00751 (visited on 11/24/2023).

[20] A. Aghajanyan, L. Zettlemoyer, and S. Gupta, *Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning*, Dec. 2020. [Online]. Available: http://arxiv.org/abs/2012.13255 (visited on 11/17/2023).

[21] A. Ansuini, A. Laio, J. H. Macke, and D. Zoccolan, *Intrinsic dimension of data representations in deep neural networks*, Oct. 2019. DOI: 10.48550/arXiv.1905.12784. [Online]. Available: http://arxiv.org/abs/1905.12784 (visited on 11/17/2023).

[22] G. H. Golub and C. Reinsch, "Singular Value Decomposition and Least Squares Solutions," in *Handbook for Automatic Computation: Volume II: Linear Algebra*, F. L. Bauer, A. S. Householder, F. W. J. Olver, H. Rutishauser, K. Samelson, and E. Stiefel, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1971, pp. 134–151, ISBN: 978-3-642-86940-2. [Online]. Available: https://doi.org/10.1007/978-3-642-86940-2_10.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal Policy Optimization Algorithms*, Aug. 2017. [Online]. Available: http://arxiv.org/abs/1707.06347 (visited on 11/20/2023).

[24] H. Lee, S. Phatale, H. Mansoor, *et al.*, *RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback*, arXiv:2309.00267 [cs], Nov. 2023. DOI: 10.48550/arXiv.2309.00267. [Online]. Available: http://arxiv.org/abs/2309.00267 (visited on 01/11/2024).

[25] C. Guo, J. Tang, W. Hu, *et al.*, "OliVe: Accelerating Large Language Models via Hardware-friendly Outlier-Victim Pair Quantization," en, in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, Orlando FL USA: ACM, Jun. 2023, pp. 1–15, ISBN: 9798400700958. DOI: 10.1145/3579371.3589038. [Online]. Available: https://dl.acm.org/doi/10.1145/3579371.3589038 (visited on 11/17/2023).

[26] P. Micikevicius, S. Narang, J. Alben, *et al.*, *Mixed Precision Training*, arXiv:1710.03740 [cs, stat], Feb. 2018. [Online]. Available: http://arxiv.org/abs/1710.03740 (visited on 11/17/2023).

[27] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, *QLoRA: Efficient Finetuning of Quantized LLMs*, May 2023. [Online]. Available: http://arxiv.org/abs/2305.14314 (visited on 11/22/2023).

[28] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," en, *Nature Medicine*, vol. 29, no. 8, pp. 1930–1940, Aug. 2023, Number: 8 Publisher: Nature Publishing Group, ISSN: 1546-170X. DOI: 10.1038/s41591-023-02448-8. [Online]. Available: https://www.nature.com/articles/s41591-023-02448-8 (visited on 10/07/2023).

[29] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020, ISSN: 1566-2535. DOI: 10.1016/j.inffus.2019.12.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103 (visited on 01/18/2024).

[30] A. Singhal, "Modern Information Retrieval: A Brief Overview," en, 2001.

[31] M. E. Maron and J. L. Kuhns, "On Relevance, Probabilistic Indexing and Information Retrieval," en, *Journal of the ACM*, vol. 7, no. 3, pp. 216–244, Jul. 1960, ISSN: 0004-5411, 1557-735X. DOI: 10.1145/321033.321035. [Online]. Available: https://dl.acm.org/doi/10.1145/321033.321035 (visited on 11/13/2023).

[32] N. Fuhr, "Probabilistic Models in Information Retrieval," en, *The Computer Journal*, vol. 35, no. 3, pp. 243–255, Jun. 1992, ISSN: 0010-4620, 1460-2067. DOI: 10.1093/comjnl/35.3.243. [Online]. Available: https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/35.3.243 (visited on 11/13/2023).

[33] H. Turtle and W. B. Croft, "Evaluation of an inference network-based retrieval model," en, *ACM Transactions on Information Systems*, vol. 9, no. 3, pp. 187–222, Jul. 1991, ISSN: 1046-8188, 1558-2868. DOI: 10.1145/125187.125188. [Online]. Available: https://dl.acm.org/doi/10.1145/125187.125188 (visited on 12/04/2023).

[34] T.-Y. Liu, "Learning to Rank for Information Retrieval," en, *Foundations and Trends® in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2007, ISSN: 1554-0669, 1554-0677. DOI: 10.1561/1500000016. [Online]. Available: http://www.nowpublishers.com/article/Details/INR-016 (visited on 11/16/2023).

[35] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," en, in *Proceedings of the 24th international conference on Machine learning*, Corvalis Oregon USA: ACM, Jun. 2007, pp. 129–136, ISBN: 978-1-59593-793-3. DOI: 10.1145/1273496.1273513. [Online]. Available: https://dl.acm.org/doi/10.1145/1273496.1273513 (visited on 11/16/2023).

[36] B. Mitra and N. Craswell, "An Introduction to Neural Information Retrieval t," en, *Foundations and Trends® in Information Retrieval*, vol. 13, no. 1, pp. 1–126, 2018, ISSN: 1554-0669, 1554-0677. DOI: 10.1561/1500000061. [Online]. Available: http://www.nowpublishers.com/article/Details/INR-061 (visited on 11/16/2023).

[37] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 9459–9474. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html (visited on 11/16/2023).

[38] A. Toma, S. Senkaiahliyan, P. R. Lawler, B. Rubin, and B. Wang, "Generative AI could revolutionize health care — but not if control is ceded to big tech," en, *Nature*, vol. 624, no. 7990, pp. 36–38, Dec. 2023. DOI: 10.1038/d41586-023-03803-y. [Online]. Available: https://www.nature.com/articles/d41586-023-03803-y (visited on 12/10/2023).

[39] Z. Chen, A. H. Cano, A. Romanou, *et al.*, *MEDITRON-70B: Scaling Medical Pretraining for Large Language Models*, Nov. 2023. DOI: 10.48550/arXiv.2311.16079. [Online]. Available: http://arxiv.org/abs/2311.16079 (visited on 11/29/2023).

[40] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, and Y. Zhang, *ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge*, Jun. 2023. DOI: 10.48550/arXiv.2303.14070. [Online]. Available: http://arxiv.org/abs/2303.14070 (visited on 11/29/2023).

[41] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, *BERTScore: Evaluating Text Generation with BERT*, Feb. 2020. DOI: 10.48550/arXiv.1904.09675. [Online]. Available: http://arxiv.org/abs/1904.09675 (visited on 01/25/2024).

[42] T. Han, L. C. Adams, J.-M. Papaioannou, *et al.*, *MedAlpaca – An Open-Source Collection of Medical Conversational AI Models and Training Data*, Oct. 2023. [Online]. Available: http://arxiv.org/abs/2304.08247 (visited on 11/29/2023).

[43] R. Taori, I. Gulrajani, T. Zhang, *et al.*, *Stanford alpaca: An instruction-following llama model*, 2023. [Online]. Available: https://github.com/tatsu-lab/stanford_alpaca.

[44] C. Yuan, P. B. Ryan, C. Ta, *et al.*, "Criteria2Query: A natural language interface to clinical databases for cohort definition," *Journal of the American Medical Informatics Association*, vol. 26, no. 4, pp. 294–305, Apr. 2019, ISSN: 1527-974X. DOI: 10.1093/jamia/ocy178. [Online]. Available: https://doi.org/10.1093/jamia/ocy178 (visited on 11/09/2023).

[45] C. Wong, S. Zhang, Y. Gu, *et al.*, *Scaling Clinical Trial Matching Using Large Language Models: A Case Study in Oncology*, Aug. 2023. DOI: 10.48550/arXiv.2308.02180. [Online]. Available: http://arxiv.org/abs/2308.02180 (visited on 09/26/2023).

[46] R. Pradeep, Y. Li, Y. Wang, and J. Lin, "Neural Query Synthesis and Domain-Specific Ranking Templates for Multi-Stage Clinical Trial Matching," en, in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Madrid Spain: ACM, Jul. 2022, pp. 2325–2330, ISBN: 978-1-4503-8732-3. DOI: 10.1145/3477495.3531853. [Online]. Available: https://dl.acm.org/doi/10.1145/3477495.3531853 (visited on 06/15/2023).

[47] B. Koopman and G. Zuccon, "A Test Collection for Matching Patients to Clinical Trials," en, in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, Pisa Italy: ACM, Jul. 2016, pp. 669–672, ISBN: 978-1-4503-4069-4. DOI: 10.1145/2911451.2914672. [Online]. Available: https://dl.acm.org/doi/10.1145/2911451.2914672 (visited on 06/17/2023).

[48] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '09, New York, NY, USA: Association for Computing Machinery, Jul. 2009, pp. 758–759, ISBN: 978-1-60558-483-6. DOI: 10.1145/1571941.1572114. [Online]. Available: https://dl.acm.org/doi/10.1145/1571941.1572114 (visited on 11/07/2023).

[49] R. Pradeep, R. Nogueira, and J. Lin, *The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models*, Jan. 2021. DOI: 10.48550/arXiv.2101.05667. [Online]. Available: http://arxiv.org/abs/2101.05667 (visited on 11/07/2023).

[50] R. Nogueira, Z. Jiang, and J. Lin, *Document Ranking with a Pretrained Sequence-to-Sequence Model*, en, arXiv:2003.06713 [cs], Mar. 2020. [Online]. Available: http://arxiv.org/abs/2003.06713 (visited on 10/10/2023).

[51] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," en, 2020.

[52] Q. Jin, C. Tan, Z. Zhao, Z. Yuan, and S. Huang, "Alibaba DAMO Academy at TREC Clinical Trials 2021: Exploring Embedding-based First-stage Retrieval with TrialMatcher," en, 2021.

[53] E. Alsentzer, J. Murphy, W. Boag, *et al.*, "Publicly Available Clinical BERT Embeddings," in *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, A. Rumshisky, K. Roberts, S. Bethard, and T. Naumann, Eds., Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 72–78. DOI: 10.18653/v1/W19-1909. [Online]. Available: https://aclanthology.org/W19-1909 (visited on 11/07/2023).

[54] J. Gao, C. Xiao, L. M. Glass, and J. Sun, "COMPOSE: Cross-Modal Pseudo-Siamese Network for Patient Trial Matching," en, in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Virtual Event CA USA: ACM, Aug. 2020, pp. 803–812, ISBN: 978-1-4503-7998-4. DOI: 10.1145/3394486.3403123. [Online]. Available: https://dl.acm.org/doi/10.1145/3394486.3403123 (visited on 07/06/2023).

[55] R. K. Srivastava, K. Greff, and J. Schmidhuber, *Highway Networks*, Nov. 2015. DOI: 10.48550/arXiv.1505.00387. [Online]. Available: http://arxiv.org/abs/1505.00387 (visited on 11/08/2023).

[56] J. Weston, S. Chopra, and A. Bordes, *Memory Networks*, en, Nov. 2015. [Online]. Available: http://arxiv.org/abs/1410.3916 (visited on 06/29/2023).

[57] X. Zhang, C. Xiao, L. M. Glass, and J. Sun, *DeepEnroll: Patient-Trial Matching with Deep Embedding and Entailment Prediction*, Jan. 2020. [Online]. Available: http://arxiv.org/abs/2001.08179 (visited on 07/07/2023).

[58] B. P. Theodorou, C. Xiao, and J. Sun, "TREEMENT: Interpretable Patient-Trial Matching via Personalized Dynamic Tree-based Memory Network," en, in *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, Houston TX USA: ACM, Sep. 2023, pp. 1–9, ISBN: 9798400701269. DOI: 10.1145/3584371.3612998. [Online]. Available: https://dl.acm.org/doi/10.1145/3584371.3612998 (visited on 11/09/2023).

[59] Q. Jin, Z. Wang, C. S. Floudas, J. Sun, and Z. Lu, *Matching Patients to Clinical Trials with Large Language Models*, Jul. 2023. DOI: 10.48550/arXiv.2307.15051. [Online]. Available: http://arxiv.org/abs/2307.15051 (visited on 09/26/2023).

[60] D. M. d. Hamer, P. Schoor, T. B. Polak, and D. Kapitan, *Improving Patient Pre-screening for Clinical Trials: Assisting Physicians with Large Language Models*, Jun. 2023. DOI: 10.48550/arXiv.2304.07396. [Online]. Available: http://arxiv.org/abs/2304.07396 (visited on 07/12/2023).

[61] G. Peikos, S. Symeonidis, P. Kasela, and G. Pasi, *Utilizing ChatGPT to Enhance Clinical Trial Enrollment*, Jun. 2023. [Online]. Available: http://arxiv.org/abs/2306.02077 (visited on 11/07/2023).

[62] J. Yuan, R. Tang, X. Jiang, and X. Hu, *LLM for Patient-Trial Matching: Privacy-Aware Data Augmentation Towards Better Performance and Generalizability*, Mar. 2023. DOI: 10.48550/arXiv.2303.16756. [Online]. Available: http://arxiv.org/abs/2303.16756 (visited on 06/24/2023).

[63] T. Brown, B. Mann, N. Ryder, *et al.*, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html (visited on 11/09/2023).

[64] E. Saravia, *Prompt Engineering Guide*, Publication Title: https://github.com/dair-ai/Prompt-Engineering-Guide original-date: 2022-12-16T16:04:50Z, Dec. 2022. [Online]. Available: https://github.com/dair-ai/Prompt-Engineering-Guide (visited on 02/25/2024).

[65] M. Lewis, Y. Liu, N. Goyal, *et al.*, *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*, Oct. 2019. DOI: 10.48550/arXiv.1910.13461. [Online]. Available: http://arxiv.org/abs/1910.13461 (visited on 01/25/2024).

[66] Z. Li, X. Li, Y. Liu, *et al.*, *Label Supervised LLaMA Finetuning*, Oct. 2023. [Online]. Available: http://arxiv.org/abs/2310.01208 (visited on 11/25/2023).

[67] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, pp. 37–46, 1960, ISSN: 1552-3888. DOI: 10.1177/001316446002000104.

Appendix

A. One-Shot Example

A.1. Example *patient-trial pair* used for both tasks

Here is an example patient note: A 46-year-old Asian woman with MDD complains of persistent feelings of sadness and loss of interest in daily activities. She states that her mood is still depressed most of the days. She complains of loss of energy and feelings of worthlessness nearly every day. She is on antidepressants for the past 6 months, but the symptoms are still present. She does not drink alcohol or smoke. She used to exercise every day for at least 30 min., but she doesn't have enough energy to do so for the past 3 weeks. She also has some digestive issues recently. She is married and has 4 children. She is menopausal. Her husband was diagnosed with colon cancer a year ago and is undergoing chemotherapy. Her past medical history is unremarkable. Her family history is negative for any psychologic problems. Her HAM-D score is 20.

Here is an example clinical trial: Title: Theta Burst Study Ghent\: 50 Right-handed antidepressant-free unipolar depressed patients (age 18-65 years) will be treated with in total 20 Theta burst sessions; these will be spread over 4 days. On each stimulation day, a given patient shall receive 5 sessions with a between session delay of 10 to 15 minutes. Patients will be selected using the structured Mini-International Neuropsychiatric Interview (MINI). All will be at least stage I treatment resistant. Because concomitant antidepressant treatment can confound outcome results, all patients will go through a medication washout before entering the study and they will be free from any antidepressant, neuroleptic and mood stabilizer for at least two weeks before entering the treatment protocol. Only habitual benzodiazepine agents will be allowed. All patients will be closely matched for gender and age with 50 never-depressed medication-free healthy volunteers. Only baseline measurements will be collected: no volunteer will undergo the treatment. We expect that real Theta burst treatment and not sham will result in a significant and clinical meaningful response. Intervention Type: Other *Inclusion Criteria:* - antidepressant-free unipolar depression - age between 18 and 65 years - right handed - at least stage I treatment resistant Healthy volunteers:Accepts Healthy Volunteers *Exclusion*

Criteria: - current or past history of epilepsy, neurosurgical interventions, having a pacemaker or metal or magnetic objects in the brain, alcohol dependence and suicide attempts within 6 months before the start of the study.

A.2. Example *response* for the classification task

Here is an example response: "A: eligible"

A.3. Example *response* for the reasoning task

Here is an example response: *Inclusion Criteria*

- antidepressant-free unipolar depression: Eligible
- age between 18 and 65 years: Eligible
- right-handed: Not Relevant for Clinical Trial
- at least stage I treatment resistant: Eligible *Exclusion Criteria:*
- current or past history of epilepsy: Not Relevant for Clinical Trial
- neurosurgical interventions: Not Relevant for Clinical Trial
- having a pacemaker: Not Relevant for Clinical Trial\nmetal or magnetic objects in the brain: Not Relevant for Clinical Trial
- alcohol dependence: Eligible
- suicide attempts within 6 months before the start of the study: Eligible

Based on the provided information and the criteria for the clinical trial, the patient would likely be considered eligible for the clinical trial since she meets the inclusion criteria and does not meet the specified exclusion criteria. Concluding, the patient is: "A: eligible"

B. *trec_eval* Script Output Metrics

```
runid                   all     dqducscw-340_v7_None_4_v5
num_q                   all     8
num_ret                 all     1000
num_rel                 all     1552
num_rel_ret             all     380
map                     all     0.1637
gm_map                  all     0.1518
Rprec                   all     0.2300
bpref                   all     0.2085
recip_rank              all     1.0000
iprec_at_recall_0.00    all     1.0000
iprec_at_recall_0.10    all     0.6997
iprec_at_recall_0.20    all     0.5205
iprec_at_recall_0.30    all     0.0000
iprec_at_recall_0.40    all     0.0000
iprec_at_recall_0.50    all     0.0000
iprec_at_recall_0.60    all     0.0000
iprec_at_recall_0.70    all     0.0000
iprec_at_recall_0.80    all     0.0000
iprec_at_recall_0.90    all     0.0000
iprec_at_recall_1.00    all     0.0000
P_5                     all     0.7000
P_10                    all     0.7375
P_15                    all     0.7083
P_20                    all     0.6813
P_30                    all     0.6708
P_100                   all     0.4463
P_200                   all     0.2375
P_500                   all     0.0950
P_1000                  all     0.0475
```

Figure B.1.: *trec_eval* script run *TrialLlama* on 2021 dataset.

```
runid                   all     dqducscw-340_v7_None_4_v5
num_q                   all     4
num_ret                 all     1000
num_rel                 all     294
num_rel_ret             all     128
map                     all     0.1501
gm_map                  all     0.1176
Rprec                   all     0.2449
bpref                   all     0.1917
recip_rank              all     0.5952
iprec_at_recall_0.00    all     0.6487
iprec_at_recall_0.10    all     0.4978
iprec_at_recall_0.20    all     0.3683
iprec_at_recall_0.30    all     0.2708
iprec_at_recall_0.40    all     0.1613
iprec_at_recall_0.50    all     0.0000
iprec_at_recall_0.60    all     0.0000
iprec_at_recall_0.70    all     0.0000
iprec_at_recall_0.80    all     0.0000
iprec_at_recall_0.90    all     0.0000
iprec_at_recall_1.00    all     0.0000
P_5                     all     0.4500
P_10                    all     0.3750
P_15                    all     0.4000
P_20                    all     0.3625
P_30                    all     0.3500
P_100                   all     0.2400
P_200                   all     0.1475
P_500                   all     0.0640
P_1000                  all     0.0320
```

Figure B.2.: *trec_eval* script run *TrialLlama* on 2022 dataset.

C. Response Examples

C.1. Example 1

This example shows that *TrialLlama* understands the underlying data well. Nevertheless, it struggles between inclusion and exclusion criteria, as seen in the verdict. Even tough it is right, the *TrialLlama* did not mention, that the BMI of the patient is too high, meeting an exclusion criterion.

TrialLlama response

1. Does the patient have moderate to severe obstructive sleep apnea? No, the patient does not have moderate to severe obstructive sleep apnea. The patient is complaining of sleep apnea, but the severity is not specified.
2. Does the patient have a BMI of 40 kg/m2 or higher? Yes, the patient has a BMI of 41.54, which is higher than 40 kg/m2.
3. Is the patient aged 30-75 years? Yes, the patient is aged 34 years, which is within the range of 30-75 years.
4. Does the patient have a vitamin D2D3 level $<$ 30 ng/ml? No, the patient's vitamin D2D3 level is not specified. Based on the information provided, the patient is not eligible for the clinical trial. The patient does not have moderate to severe obstructive sleep apnea, and their vitamin D2D3 level is not specified.

Patient topic

The patient is a 34-year-old obese woman who comes to the clinic with weight concerns. She is 165 cm tall, and her weight is 113 kg (BMI: 41.5). In the past, she unsuccessfully used antiobesity agents and appetite suppressants. She is complaining of sleep apnea, PCO and dissatisfaction with her body shape. She is a high-school teacher. She is married for 5 years. She doesn't use any contraceptive methods for the past 4 months and she had no prior pregnancies. She rarely exercises and movement seems to be hard for her. She is not able to complete the four-square step test in less than 15 seconds. She does not smoke or use any drugs. Her BP: 130/80, HR: 195/min and her BMI is: 41.54. Her lab results: FBS: 98 mg/dl TG: 150 mg/dI Cholesterol: 180 mg/di

CT

Title: Sleep Apnea Syndrome and Vitamin D
Summary: Obstructive sleep apnea/hypopnea syndrome (OSAHS) is a highly prevalent disorder with multiple comorbidities. OSAHS is characterized by repetitive episodes of airflow reduction (hypopnoea) or cessation (apnoea) due to upper airway collapse during sleep. Its major risk factor is obesity. However, its pathogenesis is complex and multifactorial. Reduced upper airway muscle tonus and/or unstable neuromuscular output seem to be involved in this collapsus. A normal vitamin D status is necessary for normal muscle

function and neuromuscular output. As obesity is associated with a high rate of hypovitaminosis D, it appears of interest to evaluate the effect of vitamin D supplementation on OSAHS patients with vitamin D deficiency.
Intervention Type: Drug
INCLUSION CRITERIA:- Moderate to severe obstructive sleep apnoea (AHI > 15) - With or without continuous positive airway pressure for a minimum of six months - aged 30-75 years - Vitamin D2D3 level < 30 ng/ml
EXCLUSION CRITERIA: - Diseases or therapy known to interfere with phosphocalcium or vitamin D metabolism - Patients with mixed sleep apnoea (central and obstructive) - Patients with CPAP use < 4 hours per night - Patients with maxillofacial or oro-pharyngeal diseases) - Patients with chronic respiratory failure, hypercapnic patients - Muscle diseases - Alcohol intake > 2 glasses per day - Body mass index > 40.

Ground Truth

Ground Truth: Excluded

C.2. Example 2

This example shows the model's focus on inclusion criteria. Nevertheless, in this case, all exclusion criteria were simply put under point nr. 4.

TrialLlama response

1. Does the patient meet the BMI criteria? BMI = 41.54 (calculated from 113 kg / 165 cm) BMI 30 kg/m2: Yes
2. Does the patient have any of the comorbidities listed in the inclusion criteria? hypertension: Not mentioned dyslipidemia: Not mentioned obstructive sleep apnea: Yes cardiovascular disease: Not mentioned
3. Has the patient had at least one unsuccessful dietary effort to lose body weight? Yes (mentioned in the patient note)
4. Does the patient have any exclusion criteria? diabetes mellitus: Not mentioned obesity induced by other endocrinologic disorders or monogenetic or syndromic forms of obesity: Not mentioned history of pancreatitis: Not mentioned family or personal history of medullary thyroid carcinoma (MTC) or multiple endocrine neoplasia syndrome type 2 (MEN-2): Not mentioned history of significant active or unstable major depressive disorder (MDD) or other severe psychiatric disorder within the last 2 years: Not mentioned Any lifetime history of a suicide attempt: Not mentioned Based on the information provided, the patient is eligible for the clinical trial.

Patient topic

The patient is a 32-year-old obese woman who came to the clinic with weight concerns. She is 165 cm tall and her weight is 113 kg. She is complaining of sleep apnea, PCO and dissatisfaction with her body shape. She is a high-school teacher married for 5 years.

She doesn't use any contraceptive methods for the past 4 months and she had no prior pregnancies. She doesn't smoke or use any drugs. She likes to try diets and exercise to lose weight. She completed the four square step test in 14 seconds. Her BP: 130/80, HR: 195/min and her BMI is: 41.54. Her labs: FBS: 98 mg/dl TG: 150 mg/dl Cholesterol: 180 mg/d LDL: 90 mg/dl HDL: 35 mg/dl Her cardiac assessment is normal. Her joints and ROM are within normal.

CT

Title: A Study of Tirzepatide (LY3298176) in Participants With Obesity or Overweight for the Maintenance of Weight Loss
Summary: This is a study of tirzepatide in participants with obesity or overweight. The main purpose is to learn more about how tirzepatide maintains body weight loss. The study has two phases: a lead-in phase in which all participants take tirzepatide and a treatment phase in which participants will either continue tirzepatide or switch to placebo. The study will last about 2 years (25 visits).
Intervention Type: Other
INCLUSION CRITERIA: - Body Mass Index (BMI) 30 kilograms per square meter (kg/m-), or 27 kg/m? and previous diagnosis with at least one of the following comorbidities: hypertension, dyslipidemia, obstructive sleep apnea, cardiovascular disease - History of at least one unsuccessful dietary effort to lose body weight
EXCLUSION CRITERIA: - Diabetes mellitus - Change in body weight greater than 5 kg within 3 months prior to starting study - Obesity induced by other endocrinologic disorders or monogenetic or syndromic forms of obesity - History of pancreatitis - Family or personal history of medullary thyroid carcinoma (MTC) or multiple endocrine neoplasia syndrome type 2 (MEN-2) - History of significant active or unstable major depressive disorder (MDD) or other severe psychiatric disorder within the last 2 years - Any lifetime history of a suicide.

Ground Truth

Ground Truth: *Eligible*

C.3. Example 3

This example highlights the models capabilities of highlighting non-provided information in the patient description. Also, the third statement shows the described problem of hallucination based on other, similar topics. The answer is addressed significant comorbidities, which were never mentioned, neither in the CT nor in the patient description. This hallucination could emerge from the small repetitiveness of patient-trial pairs described in the limitations chapter, causing the model to refer to other CTs.

TrialLlama response

1. Does the patient have a BMI 30 kg/m2? Yes, the patient has a BMI of 41.54, which is greater than 30 kg/m2, so the first inclusion criterion is met.

2. Does the patient have a high Apne-hypopnea index (AHI)? The patient does not have a high AHI, as this information is not provided in the patient note. Therefore, the second inclusion criterion is not met.

3. Does the patient have any significant comorbidities? The patient has sleep apnea, which is a significant comorbidity. However, the exclusion criterion states that patients with significant comorbidities are excluded, so the patient would not be eligible for the clinical trial. Therefore, the patient is not eligible for the clinical trial.

Patient topic

Here is the patient note: The patient is a 34-year-old obese woman who comes to the clinic with weight concerns. She is 165 cm tall, and her weight is 113 kg (BMI: 41.5). In the past, she unsuccessfully used antiobesity agents and appetite suppressants. She is complaining of sleep apnea, PCO and dissatisfaction with her body shape. She is a high-school teacher. She is married for 5 years. She doesn't use any contraceptive methods for the past 4 months and she had no prior pregnancies. She rarely exercises and movement seems to be hard for her. She is not able to complete the four-square step test in less than 15 seconds. She does not smoke or use any drugs. Her BP: 130/80, HR: 195/min and her BMI is: 41.54. Her lab results: FBS: 98 mg/d TG: 150 mg/d Cholesterol: 180 mg/d LDL: 90 mg/dl HDL: 35 mg/dl She is considering a laparoscopic gastric bypass.

CT

Title: The Effect of Aerobic Interval Training on Obstructive Sleep Apnea
Summary: The purpose of this study is to investigate if 3 months of interval training improves obstructive sleep apnea in obese patients diagnosed with moderate to severe obstructive sleep apnea. The working hypothesis is that 3 months of 3 weekly aerobic interval training sessions improve obstructive sleep apnea and sleep quality in obese patients.
Intervention Type: Behavioral
INCLUSION CRITERIA: - BMI 30 kg/M2 - Apne - hypopnea index (AHI) $>$ 10 - No significant comorbidities - Abel to exercise
EXCLUSION CRITERIA: - Inability to exercise due to musculoskeletal conditions - Known ischemic cardiovascular disease - Drug abuse - Mental illnesses.

Ground Truth

Ground Truth: *Eligible*

C.4. Example 4

This example shows how *TrialLlama* also understand non-applicable criteria, not applying to the patient's profile and correctly classifying the patient as *eligible*.

TrialLlama response

1. Is the patient 30 years or older? Yes, the patient is 55 years old, which is greater than 30 years old.
2. Does the patient have a diagnosis of Parkinson's disease? Yes, the patient has been diagnosed with Parkinson's disease.
3. Does the patient have adequate contraceptive measures (females)? Not applicable, as the patient is a male.
4. Has the patient had any prior treatment with anti-PD medication? No, the patient has not had any prior treatment with anti-PD medication.
5. Is the patient taking any medications that could interfere with the study drug? No, the patient is not taking any medications that could interfere with the study drug.
6. Does the patient have any other clinically significant medical or psychiatric conditions? No, the patient does not have any other clinically significant medical or psychiatric conditions. Therefore, the patient is eligible for the clinical trial.

Patient topic

The patient is a 55-year-old man who was recently diagnosed with Parkinson's disease. He is complaining of slowness of movement and tremors. His disease is ranked as mild, Hoehn-Yahr Stage I. His past medical history is significant for hypertension and hypercholesterolemia. He lives with his wife. They have three children. He used to be active with gardening before his diagnosis. He complains of shaking and slow movement. He had difficulty entering through a door, as he was frozen and needed guidance to step in. His handwriting is getting smaller. He is offered Levodopa and Trihexyphenidyl. He is an alert and cooperative man who does not have any signs of dementia. He does not smoke or use any illicit drugs.

CT

Title: A Trial of MitoQ for the Treatment of People With Parkinson's Disease
Summary: In Parkinson's Disease, the mitochondrial membranes in cells that produce dopamine become damaged by oxidants, leading to the death of these cells and progressive tremor, slowness of movement and the loss of neurons in the substantia nigra (a part of the brain that is involved in movement). Mitoquinone is targeted to reach the membrane of mitochondria and provide protection from damaging oxidants. There are no treatments currently available to slow the progression of PD and this trial will help advance the development of this unique disease modifying drug. This trial will enroll 120 participants with untreated early onset of PD. Participants will be randomized to receive 1 of 3 treatments: 40 mg of MitoQ tablets, 80 mg of MitoQ tablets or placebo. The researchers, participants and sponsor will all be blinded to the treatment allocation. Participants will be assessed after 1, 2, 3, 6, 9, 12 months of treatment and again 28 days after their last dose. The effectiveness of the trial drug will be measured via the Unified Parkinson's Disease Rating

Scale (UPDRS). The safety of the trial drug will be monitored via regular participant examinations, blood tests, ECG and collecting information on adverse events.
Intervention Type: Drug
INCLUSION CRITERIA: 1. Informed consent 2. 30 yrs or older 3. Diagnosis of PD (2 or more of bradykinesia; rest tremor, rigidity) 4. Adequate contraceptive measures (females)
EXCLUSION CRITERIA: 1. Malignancy within last 2 years 2. Pregnancy & breast-feeding 3. Treatment with any anti-PD drugs within 30 days of enrolment 4. Prior treatment with anti-PD medication exceeding 42 days in total 5. Medication-induced PD/PD not of idiopathic origin 6. CoQ10/idebenone doses of 300mg/day or higher within 120 days, >25mg/day within 7 days of enrolment 7. Methylphenidate HCl, neuroleptics, reserpine, amphetamines, selegeline or MAOIs within 6 months of enrolment 8. CNS medications at unstable doses within 60 days of enrolment 9. Dietary supplements > 5 x RDI 10. Hypersensitivity to CoQ10, idebenone or any components of the study drug 11. Unable to swallow 12. Diseases with features of PD 13. Seizure(s) within 12 months prior to enrolment 14. UPDRS tremor score of 4 15. Hamilton Depression Rating Scale score > 10 16. History of stroke 17. Requirement for dopaminergic drugs 18. Modified Hoehn & Yahr score > 2.5 19. History of brain surgery for Parkinson's disease 20. History of structural brain disease / congenital brain abnormality 21. History of ECT 22. Any other clinically significant medical or psychiatric condition or lab abnormality 23. Enrolment in any other pharmacological study within 30 days of enrolment

Ground Truth

Ground Truth: *Eligible*

D. Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Masterstudiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Unterschrift: [redacted] Ort, Datum: Hamburg, 28.02.2024

E. Erklärung zur Veröffentlichung

Ich bin damit einverstanden, dass meine Abschlussarbeit in den Bestand der Fachbereichsbibliothek eingestellt wird.

Unterschrift: [redacted] Ort, Datum: Hamburg, 28.02.2024