CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Large Language Models For Patient Document Summarization

A case study in applying large language models for patient document summarization conducted at Sahlgrenska University Hospital.

Master's thesis in Computer science and engineering

Albert Lund, Felix Nilsson

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Large Language Models For Patient Document Summarization

A case study in applying large language models for patient document summarization conducted at Sahlgrenska University Hospital.

Albert Lund, Felix Nilsson

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Large Language Models For Patient Document Summarization
A case study in applying large language models for patient document summarization conducted at Sahlgrenska University Hospital.
Albert Lund, Felix Nilsson

Supervisor: Bastiaan Bruinsma, Computer Science and Engineering
Advisors: Isak Barbopoulous, Västra Götalandsregionen,
Denitsa Saynova, Computer Science and Engineering
Examiner: Moa Johansson, Computer Science and Engineering

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Description of the picture on the cover page (if applicable)

Typeset in LaTeX
Gothenburg, Sweden 2024

Large Language Models For Patient Document Summarization
A case study in applying large language models for patient document summarization conducted at Sahlgrenska University Hospital.
Albert Lund
Felix Nilsson
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Reading patient documents is a time-consuming but necessary part of a doctor's duties, which is often further slowed down by poorly designed software systems. This, in turn, contributes to the already psychologically stressful environment of being a doctor. However, large language models (LLMs) have recently shown excellent results on many downstream tasks, including summarization. Moreover, performance on such tasks shows little degradation when transferred to a language other than English, despite relatively limited exposure to the target language. In this thesis, we show how LLMs can save time in healthcare by generating automatic summaries over patient document. In particular, we closely examine the potential of open-source LLMs, which allow for more control, in contrast to proprietary LLMs, which currently represent the state of the art. To this end, we design an automatic evaluation procedure that compares a given model's summarization capabilities to that of a clinician. We then optimize an open-source LLM via finetuning to show performance comparable to GPT-4 on the said procedure. Finally, we conduct a small-scale study in which doctors compare summaries produced by our LLM solution to those of a rule-based summarizer and a doctor. We find that while doctors prefer the human summary, the LLM outperformed the rule-based summarizer. Interpreting these results, we see the future of automatic medical summarization as promising. However, in our view, the use of a novel technology such as LLMs needs to be navigated carefully to avoid harming patients. The thesis was conducted at Sahlgrenska University Hospital (SU), where it was part of a larger project looking at AI in healthcare, and it was organized by SU's AI Competence Center (AICC).

Keywords: Large Language Models, Healthcare, Natural Language Processing, Machine Learning

Acknowledgements

We would like to thank Bastiaan and Denitsa for their helpful advice and constructive feedback throughout the project, which was vital to its success.

We would also like to thank Isak for designing an interesting and relevant thesis, as well as Sahlgrenska for offering resources to conduct the relevant experiments.

Finally, a special acknowledgment goes to the doctors who participated in the form and graciously donated their valuable time to this project.

Felix Nilsson & Albert Lund, Gothenburg, 2024-06-03

Contents

List of Figures **xiii**

List of Tables **xv**

1 Introduction **1**
1.1 Administrative Problems . . . 1
1.2 Swedish Healthcare Problems in an International Context . . . 2
1.3 Effects on Physicians . . . 2
1.4 Sahlgrenska University Hospital . . . 2
1.5 Problem Statement . . . 3

2 Background **5**
2.1 Historical Approaches to Summarization . . . 5
2.2 Medical Summarization . . . 6
2.3 Multilinguality & Language Transfer . . . 6
2.4 Metrics . . . 7
2.4.1 BLEU . . . 7
2.4.2 ROUGE . . . 8
2.4.3 METEOR . . . 8
2.4.4 BERTScore . . . 10
2.4.5 Perplexity . . . 11
2.4.6 Criticisms of Automatic Evaluation . . . 11
2.5 Baseline/TextRank . . . 12
2.6 Model Tailoring . . . 13
2.6.1 Prompt Engineering . . . 13
2.6.2 Hyperparameters for Generation . . . 14
2.6.2.1 Temperature . . . 14
2.6.2.2 Top-k . . . 14
2.6.2.3 Top-p . . . 15
2.6.2.4 Local typicality . . . 15
2.6.2.5 Conditional probability cutoff . . . 16
2.6.3 Training . . . 16
2.6.3.1 Low rank adaptation . . . 17
2.6.3.2 Noisy Embeddings Fine Tune . . . 17
2.7 Ethical Considerations . . . 17

2.7.1 Misguiding information . . . 18
2.8 Related Work . . . 19
2.9 Purpose . . . 20

3 Methods **21**
3.1 Data . . . 21
3.1.1 Summarization task . . . 21
3.1.2 Training . . . 22
3.2 Evaluation . . . 23
3.2.1 Form . . . 24
3.3 Model Selection . . . 25
3.4 Model Tailoring . . . 26
3.4.1 Prompt Engineering . . . 26
3.4.2 Hyperparameters . . . 27
3.4.3 Training . . . 27
3.4.3.1 Domain adaptation . . . 27
3.4.3.2 Finetuning . . . 28
3.4.3.3 Combinations . . . 28
3.5 Limitations . . . 29
3.6 Computational Resources . . . 29

4 Results **31**
4.1 Model Choice . . . 31
4.1.1 Scaling Effects . . . 32
4.2 Prompt engineering . . . 33
4.2.1 Recursive summarization . . . 34
4.3 Hyperparameter tuning . . . 34
4.4 Domain adaption & Questions and Answers . . . 35
4.5 Fine-tuning . . . 37
4.6 Expert Evaluation . . . 39
4.7 Compression . . . 42

5 Discussion **45**
5.1 Open Source vs Proprietary . . . 45
5.2 Parameters vs training . . . 46
5.3 Generated Data Commentary . . . 47
5.4 Optimization Outcome . . . 49
5.4.1 Optimization target . . . 50
5.5 Performance compared to human references and TextRank . . . 51
5.6 Answer sample . . . 53

6 Conclusion **55**
6.1 Applicability . . . 55
6.2 Limitations . . . 57

Bibliography **59**

A Appendix 1 **I**

List of Figures

3.1 A synthetic data example. Notes produced by the doctor use special conventions and acronyms to speed up the notetaking process. 22

4.1 Comparison of how scaling up Llama-2 parameter count affects performance on NLP metrics and the average time spent per summary during generation. Performance on metrics shows diminishing returns when scaling up, whereas computation time grows drastically. 33
4.2 Train and evaluation loss during domain adaptation. Evaluation loss was computed every 100th step and can be seen to decrease. 36
4.3 Training and evaluation loss of the task-based domain adaptation. Evaluation was performed every 600th step. 36
4.4 Loss during LoRA fine tuning for both training and evaluation. Evaluation was performed every 50th step and also once before and after complete training. 38
4.5 ROUGE-L, BLEU and METEOR score of the best performing models of two other projects focusing on medical summarization. Note the study by Van Veen et al., the METEOR score was not reported. . . . 38
4.6 Comparison of summary between our model and the human. For this instance our model was rated poorly and the human summary was rated well, 1 and 7 respectively in overall score. 41
4.7 Comparison of summary between our model and the human. For this instance both our model and the human summary was rated to be of high quality, 7 and 9 respectively in overall score respectively. 42

A.1 An example of chain of thought prompting. The reasoning provided in the one-shot example makes the language model behave similarly, and improve their accuracy. Source: [57] I

List of Tables

3.1 Question and Answer sample. An English translation can be found in the Appendix, Table A.2. 22
3.2 Number of samples and tokens in each data set 23
3.3 The questions used in the evaluation form. Best value column indicates which value is the most favourable for a summary on the given question. 24
3.4 The three different model compositions trained in the project. 29

4.1 Comparison of models. Higher is better across all metrics. The best score in each category is bold, $n = 24$. GPT-4 significantly outperforms current open-source models, that at best perform similarly to the level of GPT-3.5-turbo. 31
4.2 Table of hyperparameters for the generation configuration and their values. Floating point parameters were rounded down for space. A value of 1 means the truncation scheme is not used. 34
4.3 Ablation study showing the performance increase of the generation config. Performance was improved across all metrics. 35
4.4 Ablation study showing the performance decrease when domain adapting and question answers. 35
4.5 Ablation study showing which model configuration performed the best. The last row displays the difference applying the best hyperparameter setting as well. The settings of hyperparameters can be found in Table 4.2 and the model configurations can be seen in Table 3.4. The performance of the Llama-2-13b base model is also included as comparison. 37
4.6 Mean of Overall scores from the evaluation. 39
4.7 Mean of Relevance scores from the evaluation. 39
4.8 Mean scores of False information present in the summaries. 39
4.9 Mean scores of Missed information from the documents. 40
4.10 Mean scores of the summaries Usefulness. 40
4.11 Mean scores of the models' proficiency in handling Medical Conventions. 40
4.12 Comparison of token lengths of the original documents, reference summaries and model output. 43

5.1 Mean scores of human summaries in evaluation 51

A.1 Summary sample II
A.2 Question and Answer sample translated to English. II
A.3 Comparison of all models and variants. Higher is better across all metrics, best score is bold, $n = 24$. III
A.4 Table showing the difference in performance between the best and the worst hyperparameter configurations, and that they correlate with BERTScore. III

1

Introduction

In healthcare, the time available to doctors and nurses needs to be allocated carefully so that patients get the treatment they need. However, this is not always the case as significant time in practice is spent navigating cumbersome information systems. Such tasks include reading a patient's medical history or parsing tables for a specific piece of information.

1.1 Administrative Problems

A study from 2023 on the topic of unnecessary administration in healthcare in Sweden supports this view [1]. Several different anecdotes tell of information systems that are not designed to be efficient with the time of medical professionals. One in particular, an anonymous anesthesiologist, notes a journal system that was designed to need several mandatory fields to be filled out. While these may be important for consistency, each field adds a time penalty each time a new patient is registered. The study goes on to name "digital anamnesis", the medical history of a patient as produced by a computer algorithm, as a low-hanging fruit among potential areas to be digitized.

A study by McKinsey on productivity in the world of healthcare from 2022 showed similar results [2]. Physicians reportedly spend an average of 19% of their time, equivalent to one workday per regular workweek, on administration alone. Moreover, slow and poorly designed information systems are named key time wasters. One participant estimated that they spent around 30 minutes per day just logging into different systems.

Nevertheless, administration problems are not only caused by information systems. The study finds that problems also stem from a clear lack of roles, where doctors, in many cases, have to perform duties beyond what is typically expected. While tasks like cleaning lunch areas or making coffee may be simple, the time it takes to complete them takes away from what could otherwise be spent with patients.

Another such issue is that the number of nurses has increased at a slower rate compared to doctors. In Sweden, there are approximately 2.7 nurses per doctor which is a low figure compared to the 4.6 nurses per doctor in countries like Denmark and Finland. This further contributes to the burden of work put on doctors, diluting their already busy schedules.

1.2 Swedish Healthcare Problems in an International Context

Healthcare systems can vary significantly between countries, and problems found in one may not be present everywhere. In Sweden, the most common type of given care is called primary care (*primärvård*) and is carried out by primary physicians. It serves as the initial point of contact, where a first assessment is made and either care is given directly or alternatively the patient is given a remittance to a specialist. It includes all types of care that do not require the medicinal or technical resources of a hospital [3].

In 2022, *the International Health Policy Survey* (IHP) demonstrated how Swedish healthcare differs when compared to the international community. This was conducted by surveying 6000 primary physicians [4]. One notable result was the lack of continuity. In other countries, it is common to use a system in which a patient has a family doctor with whom they will meet many times and form a relationship. Only one-third of Swedish citizens have such a doctor, which leads to Swedish primary physicians having longer meetings and seeing fewer patients per hour. This stems from the need to constantly read up on patient backgrounds from scratch and overall lower familiarity with the patients.

While primary care is meant to help patients find the specialist care they need, information transfer between the two is not only cited as a problem but something that has worsened over time. Physicians say it is rare to be notified of a patient's release from the hospital within 24 hours, despite this being a legal requirement. This represents a deviation from the international community, where doctors are generally more satisfied with the flow of information between healthcare instances.

1.3 Effects on Physicians

Working in an environment where productivity is constantly hamstrung by administration and inefficient IT systems not only worsens the quality of care but also affects the mental health of physicians. Swedish primary physicians are among the most stressed in the world and 2 out of 3 doctors consider their work as "incredibly stressful"[4]. Only half of physicians say their work is satisfying, and almost one-third are considering changing their careers within the next three years.

As pointed out previously, the effects are not only stemming from inefficiency but also from poor allocation of the physicians time. A study from 2012 showed that doctors who perceived themselves to have "illegitimate tasks" (*oskäliga uppgifter*) were five times as likely to be at risk for exhaustion [5].

1.4 Sahlgrenska University Hospital

Sahlgrenska University (SU) Hospital is the largest hospital in Sweden, both in terms of employees (17000) and places of care (*vårdplatser*) (2300). It is located

in the city of Gothenburg and is part of the region of Västra Götaland (*Västra Götalandsregionen*, VGR). In total, it provides care to the 1 million citizens in and around Gothenburg [6][7].

Moreover, Sahlgrenska also conducts research in different areas of healthcare and medicine. In 2021, AI Competence Center (AICC) was founded as a vehicle to explore the use of AI in healthcare and has, for example, conducted research in the area of AI as an aid in decisionmaking (*beslutstöd*) [8]. This thesis was part of a larger project conducted at AICC which aimed to research the potential use of language models in healthcare in order to reduce the administrative problems which are present there.

At SU, one example detailed a surgical unit consisting of three nurses. On average, one of the three nurses had to spend an entire day, every day, navigating such systems. This is done to collect information necessary for the next day's procedures but effectively reduces the workforce by one-third. Cases like this are seemingly the rule, not the exception.

1.5 Problem Statement

Today, much of the time that could be spent by doctors and clinicians[1] in general interacting with patients is wasted on inefficiencies within the healthcare system. While that is in part due to the assignment of duties that lie outside the definitions of their roles, a significant portion of time is also wasted on an information-gathering process that is slowed down by poorly designed IT systems. This problem is worsened by factors like the information-gathering process being more extensive and having fewer nurses per doctor when compared internationally. Ultimately this directly affects both the productivity and the level of stress experienced by doctors.

The goal of this project was to reduce this time waste by applying large language models (LLMs) to the problem of information gathering. LLMs, such as GPT-4, have shown impressive abilities in many benchmarks, often eliminating the need for human intervention in several downstream tasks [9].

Sahlgrenska's end goal is to have a chatbot in place to aid employees in retrieving, compiling, and evaluating information in their systems. This project focused on the specific subtask of summarizing patient's medical notes from their previous visits. Then, the summaries can either be used directly by the staff or provided as additional information to Sahlgrenska's chatbot.

[1]The role of clinician includes any profession working with patients directly, for example doctors, nurses and therapists.

2

Background

The purpose of a summary is to shorten a text in such a way that the new text still conveys the information contained in the original text. A summary can be considered complete if all main topics of the input are included, no redundant text is present, no text is repeated, and it is readable for the end-user [10]. However, it is not straightforward how such a summary should be constructed.

The two main methods for constructing a summary are extractive and abstractive [10]. Extractive summaries are produced by compiling the most important sentences from the corpus into a shorter text. Lead-3 is a common extractive baseline that simply identifies the first three sentences of a document and uses them as a summary [11]. Abstractive summaries use an intermediate representation of the text and generate new sentences from this source. Additionally, hybrid solutions exist that make use of both approaches.

2.1 Historical Approaches to Summarization

The earliest research in the field is credited to Luhn who uses word frequencies and their relative position in sentences to identify sentences for extraction to create abstracts of magazine articles [12]. Early adaptions of abstractive summarization concerned headline generation of news articles. One such study was performed by Banko et al. who were inspired by an IBM translation model that utilized word alignment to compute the probabilities of English words and their positions given a foreign sentence [13]. Using a corpus of headline-article pairs, they translate articles from a verbose language to the compressed language of headlines [14]. However, most research up until 2015 focused on extractive summarization, as semantic representation and natural language generation were deemed more difficult than identification for extraction [10].

Rush et al. released the first study for a fully data-driven approach for abstractive summarization using an encoder-decoder architecture composed of attention-based feed-forward neural networks [15]. Although the model only performed summarization at sentence level, the improvements still sparked the community's interest. This was further improved upon with the introduction of the transformer in 2017 [16], which gave rise to new methods within the field, resulting in a surge of popularity in research and advancements. The BART model, published in 2019, which makes use of the transformer architecture, pushed the state-of-the-art (SOTA) of summa-

rization in terms of ROUGE score (see 2.4.2) [17]. An example of a hybrid solution was explored by Hsu et al. in 2018, who used a sentence level extractor to scale the word level attention scores in an abstractive summarizer [18]. However, the current SOTA for summarization are LLMs [19] [20].

2.2 Medical Summarization

The approach of automatic text summarization has also been widely studied in the medical domain [21]. Compared to news-based datasets as XSum, CNN/DM and Multi-News, often used for summarization tasks, medical texts pose additional complexity through specific terminology, and the sensitivity of the domain constitutes further challenges for summarization [22] [23] [24]. Due to this sensitive nature, medical professionals should be able to trace the origin of information present in summaries to prevent the generation of false information. One of the earlier works dates back to 1998 when McKeown et al. created summaries of medical articles based on patient characteristics [25]. Since then, numerous techniques have been employed for the task, from rule-based systems to deep learning [21]. Due to the SOTA performance achieved by LLMs in regular text summarization, researchers have recognized the possibility of applying them to the medical domain as well [20]. For specific medical summarization tasks, it has been proven LLMs can outperform humans after adapting the model to the task [26].

2.3 Multilinguality & Language Transfer

Medical notes at Sahlgrenska are recorded in Swedish whereas most top LLMs are designed with English as their primary language. Their performances are measured with mostly English benchmarks, and the training data is mostly comprised of English. Hence, it is not straightforward whether this new SOTA technique also applies to the Swedish medical domain. For example, in both cases of the GPT-3 and Llama-2 models, English makes up close to 90% of the training dataset [27][28]. For this reason, it is important to study how well a model's performance transfers to another language, which is only a fraction as frequent in the dataset.

A study by Holmström et al. compares the performance of GPT-3 to GPT-SW3 on Swedish tasks [29]. While GPT-SW3 showed better results on perplexity (see 2.4.5) due to Swedish being significantly more represented in its training dataset, this did not translate to other tasks. For example on SweWiC, a Swedish word-in-context benchmark, GPT-SW3s performance did not exceed that of the random baseline of 50% accuracy, while GPT-3 achieved an accuracy of 61%. This suggests that even limited exposure to a language may be enough to perform well on tasks and, moreover, indicates that factors like the number of parameters and general size of the training set are perhaps more important.

Recent work by Kew et al. also shows the potential of multilingual finetuning for increased performance on downstream cross-lingual tasks [30]. In particular, instruction-tuning on a dataset with only two other languages improved helpfulness

significantly. Also relevant to our project is their examination of how multilingual instruction-tuning affected extractive question answering. Using Llama-2-7b, they found that it resulted in small but noticeable performance gains in other languages while also not degrading English performance.

2.4 Metrics

To measure the success of a generated output, it can be compared to a reference summary. The reference summary is often written by a human and is seen as a gold standard, and the best model output would exactly match the reference summary. However, such is seldom the case, and there is a need to measure the similarity between the model output and the gold standard. Four methods for the task are presented below: Bilingual Evaluation Understudy (BLEU), Recall-Oriented Understudy for Gisting Evaluation (ROUGE), Metric for Evaluation of Translation with Explicit ORdering (METEOR) and BERTScore. Additionally, the metric of perplexity is presented, which does not measure input compared to output but rather a language model's familiarity with a piece of text.

2.4.1 BLEU

BLEU was originally proposed as a metric for evaluating machine translation in 2002 [31]. However, as it uses a proposal and reference string to measure their similarity, it has also been used in other NLP tasks like summarization.

The basis of the BLEU metric lies in modified n-gram precision, p_n. For a single candidate string $\hat{x}$ and a reference string x, it is defined as.

$$p_n(\{\hat{x}\}, \{x\}) := \frac{\sum_{\text{n-gram} \in G_n(\hat{x})} \min(Count(\text{n-gram}, \hat{x}), Count(\text{n-gram}, x))}{\sum_{\text{n-gram} \in G_n(\hat{x})} Count(\text{n-gram}, \hat{x})} \tag{2.1}$$

The function $G_n(\hat{x})$ returns the set of all n-grams for the string $\hat{x}$. The function $Count(\text{n-gram}, \hat{x})$ defines the substring count of how many times the substring, in this case n-gram, appears in $\hat{x}$. Conclusively, p_n counts the number of times the n-gram appears in the candidate and the reference, takes the minimum, and sums the value over all n-grams in the candidate. The sum is then normalized by the sum of counts of n-gram in the candidate to get the score into a range of [0,1].

As the setup currently favors candidate strings shorter than the reference, a brevity penalty (BP) is introduced. Given the length of the candidate translation c, and the reference corpus length r

$$BP := \begin{cases} 1 & \text{if } c > r \\ e^{1-\frac{r}{c}} & \text{if } c \leq r \end{cases} \tag{2.2}$$

Also, the modified n-gram precision is computed up to a specific n, which in the original paper was 4. Each n-gram is then weighted by a weight w_n. In the original paper, the weights are recommended to be uniform for all n. Finally, for a candidate corpus $\hat{S}$ and a reference corpus S this gives

$$\text{BLEU}(\hat{S}, S) := \text{BP}(\hat{S}, S) \times \exp\Big(\sum_{n=1}^{N} w_n \log p_n(\hat{S}, S)\Big) \tag{2.3}$$

2.4.2 ROUGE

ROUGE is a family of metrics deliberately designed to evaluate summaries [32]. One of the metrics in the family is ROUGE-L, which makes use of the longest common subsequence (LCS) between the candidate and the reference. A sequence that is in common between the candidate $\hat{x}$ and reference x is a subset of words that appear in the same order, but not necessarily adjacent to one another, in both of the strings. LCS finds the longest of such sequences between the strings. Having found:

- the LCS
- the length of the candidate c
- the length of the reference r

The precision, recall, and F-score can be computed.

$$R_{LCS} = \frac{LCS(\hat{x}, x)}{r} \tag{2.4}$$

$$P_{LCS} = \frac{LCS(\hat{x}, x)}{c} \tag{2.5}$$

$$F_{LCS} = \frac{(1+\beta^2) R_{LCS} P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}} \tag{2.6}$$

β is a hyperparameter that adjusts the importance between precision and recall. If $\beta > 1$, recall is given more weight and vice versa. Setting $\beta = 1$ gives the harmonic mean between precision and recall, which is the standard implementation.

2.4.3 METEOR

Similarly to BLEU, METEOR was also originally designed for evaluating machine translation, in order to combat the shortcomings of BLEU [33]. Unlike BLEU, METEOR only operates on unigrams and disregards higher-order n-grams. The reasoning for doing so is introducing alignment instead.

Consider a candidate $\hat{x}$ and a reference x text. A unigram in $\hat{x}$ is aligned with a unigram in x by creating a 1:1 mapping between them. It may be aligned with a maximum of one unigram in x but could be aligned with zero. There are three

criteria that can create an alignment: exact match, stemming, and synonyms. Exact match maps unigrams only if they are equal. Stemming maps unigrams if they are equal after Porter stemming, for instance "computers" can map to "computer". Synonyms simply map to synonyms.

The alignments are created in stages, where each stage consists of two phases. In the first phase, all possible unigram matches are listed and constructed using any of the mapping schemes. In the second phase, the largest subset of the mappings is selected so that no unigram in $\hat{x}$ matches more than one unigram in x in order to construct an alignment. If there is a tie of alignments with the same number of mappings, the alignment with the fewest crosses is chosen. Given two unigram mappings (t_i, r_j) and (t_k, r_l) where t_i and t_k are from the candidate string, respectively mapped to r_j and r_l in the reference string. Making use of the position of the unigrams, the mappings create a cross if and only if

$$(\text{pos}(t_i) - \text{pos}(t_k)) \cdot (\text{pos}(r_j) - \text{pos}(r_l)) < 0 \tag{2.7}$$

These stages are repeated with the different mapping criterions to form the final alignment, starting with exact match and ending on synonyms. In each stage, only unigrams that are not already mapped are considered for mapping. Once the final alignment is constructed, the alignment precision and recall are computed.

$$P := \frac{m}{w_c} \tag{2.8}$$

$$\tag{2.9}$$

$$R := \frac{m}{w_r} \tag{2.10}$$

Here, m is the number of mappings in the final alignment from the candidate string, w_c is the number of unigrams in the candidate string, and w_r is the number of unigrams in the reference string. The precision and recall are consequently used to compute the F-score, weighing recall 9 times more than precision.

$$F := \frac{10RP}{R + 9P} \tag{2.11}$$

This score does not take into account longer n-gram matches between the strings. This is taken care of with a penalty using chunks. A chunk is a set of mapped unigrams that are adjacent both in the candidate and the reference string. The fewest possible chunks are created, and the penalty p is computed as:

$$p := 0.5 \cdot \left(\frac{\#chunks}{\#unigrams\_matched} \right)^3 \tag{2.12}$$

The final score is then computed as:

$$\text{METEOR} := F \cdot (1 - p) \tag{2.13}$$

2.4.4 BERTScore

BERTScore, developed by Zhang et al. in 2019, uses BERT to create contextual embeddings of sentences, whose similarity is then measured to produce a final score [34][35]. This approach circumvents the key reliability issue of n-gram methods, e.g. that sentences can be similar in meaning while also being worded very differently. Furthermore, it has a higher measured correlation with human judgment than previous metrics.

To compute the BERTScore between two sentences, we define the reference sentence $x = \langle x_0, \ldots, x_k \rangle$ and the candidate sentence $\hat{x} = \langle \hat{x}_0, \ldots, \hat{x}_m \rangle$ where elements of x and $\hat{x}$ are tokens. These are then both vectorized into embeddings $\mathbf{x} = \langle \mathbf{x}_0, \ldots, \mathbf{x}_k \rangle$ and $\hat{\mathbf{x}} = \langle \hat{\mathbf{x}}_0, \ldots, \hat{\mathbf{x}}_m \rangle$ respectively using an embedding model, which in this case is BERT. The embeddings are then compared token-wise using a similarity function, which here is cosine similarity. BERTScore also uses pre-normalized vectors, which simplifies the expression:

$$S_c(\mathbf{x}_i, \hat{\mathbf{x}}_j) = \frac{\mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\|\mathbf{x}_i\| \|\hat{\mathbf{x}}_j\|} = \mathbf{x}_i^\top \hat{\mathbf{x}}_j \tag{2.14}$$

For the final computation of the BERTScore, F_{BERT}, we first compute the precision by P_{BERT} matching tokens in $\hat{x}$ to x and recall R_{BERT} by matching tokens in x to $\hat{x}$:

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_i \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \qquad P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_i \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \tag{2.15}$$

$$F_{\text{BERT}} = 2 \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \tag{2.16}$$

After similarity is measured, you can optionally apply importance weighting. This step assigns more weight to rare words using inverse document frequency, which has been shown to be important when measuring similarity.

While deep-learning have dominated language modeling for some time, their applicability have only recently been extended to the area of evaluation. One such evaluation tool is G-EVAL, which outperforms other evaluation strategies by using GPT-4 to fill out a form [36].

2.4.5 Perplexity

Another metric often used to measure the performance of a language model is perplexity. Unlike the other presented metrics, BLEU, ROUGE, METEOR, and BERTScore, perplexity does not make use of reference outputs but operates directly on the conditional distributions of the language model. Given a tokenized sequence $x = \langle x_0, x_1, .., x_k \rangle$, the perplexity of the sequence is given by:

$$PPL(x) = \exp\left\{ -\frac{1}{k} \sum_{i}^{k} \log p(x_i | x_{<i}) \right\} \tag{2.17}$$

Here, $\log p(x_i|x_{<i})$ is the log-likelihood of the ith token given the preceding tokens. This is the distribution approximated by the language model. Intuitively, it can be seen as the model's measure of uncertainty. The higher the probability of the next token given the preceding tokens, the lower the perplexity. Using this metric, an estimation of a model's domain knowledge can be made.

2.4.6 Criticisms of Automatic Evaluation

The success of the automatic evaluation comes down to the success of the metrics used and the degree to which they can be manipulated. While ROUGE, BLEU, and METEOR are practical and fast for broad evaluations, they ultimately rely on simple heuristics as proxies for semantic understanding.

One major issue with ROUGE is that it treats every n-gram the same and fails to pick up on keywords that provide important context [37]. Moreover, ROUGE has been shown to be theoretically computationally hard to optimize for, and that perfect scores are impossible to attain, even in an abstractive setting [38].

While BLEU is a very commonly used metric in NLP research, it is a fairly crude measure of summarization quality. For example, it has been shown not to be able to guarantee correlation with human judgment since it does not distinguish between randomly generated permutations [39]. METEOR was designed to improve on some of the inherent weaknesses of BLEU, particularly the lack of accounting for recall [33]. For that reason, it typically correlates more with human assessments. The gold standard for semantic understanding is acknowledged in human evaluation. Hence, it is desired that the metrics correlate with human judgment.

While BERTScore improves on many flaws of older approaches, the same level of adoption in the wider research community as the other metrics mentioned has yet to be seen. Around 60% of recent papers in the field of machine translation rely on BLEU as their sole metric [40]. For recent summarization papers, around 66%, rely only on ROUGE. While there is a value in consistent use of the same metrics in the scientific community, it is also important not to ignore the flaws of BLEU, ROUGE, and METEOR.

Although the presented metrics require quality reference summaries and do not present a comprehensive value of the summaries, they are fast and accessible indica-

tors of performance. They are especially useful when comparing studies since both ROUGE and BLEU are used extensively within the research community as standard metrics for summarization.

The issue with perplexity regarding generative tasks is that it does not necessarily correlate with model performance. Muhlgay et al. show how perplexity and benchmark scores do not always agree on model ranking [41]. Likewise, when they disagree, the benchmark score is more significant determining the factuality in responses from a model. Therefore, models with a lower perplexity score within a domain could still be worse at downstream tasks compared to a more general model.

2.5 Baseline/TextRank

It is arguable whether an LLM is required when specialized summarization models already exist or Named Entity Recognition (NER) systems to classify words (medicine, diagnoses, symptoms) that could suffice as a summary. The improvements they offer might not be sufficient considering the additional challenges they impose. The main argument is the flexibility of LLMs which allows for different types of information retrieval. It is simpler to prompt an LLM for different information rather than maintaining multiple models for different purposes. Additionally, Sahlgrenska has expressed a preference in using an LLM compared to pure summarization models due to their current SOTA performance. Nevertheless, it is still useful to compare the performance to a baseline for a specific use case. Hence, the TextRank algorithm was also used in comparison to the LLMs.

The advantages of TextRank are that it is deterministic, graph-based, and extractive. Even though LLMs can be argued to have deterministic behavior using greedy text generation, the output is still inherently dependent on the prompt. The impact of the prompt on the output is not entirely trivial. Graph-based extractive algorithms tend to have better coverage than other extractive summarizers, who usually extract sentences at the beginning of documents, as it does not depend on sequence information [42]. Being an extractive algorithm is beneficial as then the summary is entirely based on the input and not on any other previously seen data. Additionally, TextRank, in particular, has shown to be competitive to some abstractive neural models in terms of ROUGE score [42].

TextRank creates a graph representing the text, with sentences as nodes. The nodes are fully connected but weighted by the number of words the sentences have in common. If the sentences have no words in common, the edge is removed. Formally, given two sentences S_j and S_i, where a sentence is represented by its sequence of words N_i. $S_i = w_1^i, w_2^i, ..., w_{N_i}^i$. The similarity between two sentences is given by:

$$s(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \cap S_j\}|}{\log(|S_i|) + \log(|S_j|)} \tag{2.18}$$

Then a modified PageRank algorithm, which allows for weigthed edges, is used to compute a node value for each node. As the original PageRank algorithm is used

on directed graphs, it makes use of the sets $\text{In}(V_i)$ to denote all vertices with an edge going into V_i and $\text{Out}(V_i)$ to denote all vertices V_i points to. In TextRank, the graph built from the similarity scores is undirected, and thus $\text{In}(V_i) = \text{Out}(V_i)$ for all i. Yet, they are still used for clarity when using undirected graphs. So, the score of vertex V_i is computed by

$$S(V_i) = (1-d) + d \cdot \sum_{V_j \in \text{In}(V_i)} \frac{w_{ji}}{\sum_{V_k \in \text{Out}(V_j)} w_{jk}} S(V_j) \tag{2.19}$$

where d is a hyperparameter recommended to be set at 0.85 and w_{ji} are the similarity scores computed between sentence (vertex) i and j. As the vertex scores are dependent on one another, the scores are computed iteratively and usually converge after 20 iterations. The vertex scores are independent of starting values, but do depend on the number of iterations. The nodes with the highest values are the sentences compiled into the extractive summarization [43].

2.6 Model Tailoring

As previously mentioned, to reliably outperform humans, general-purpose LLMs need to be tailored to fit the specific downstream task to which they are applied. The following subsections describe a few common techniques to adapt the output of LLMs.

2.6.1 Prompt Engineering

The basis of language modeling is to compute a conditional probability distribution of the next token given a sequence of preceeding tokens. Hence, the basis of text generation for an LLM comes from continuing the input provided by the user. Depending on the given input, the model will elicit specific behavior. Prompt engineering is the process of iterating over different inputs to the system in order to improve performance. One part of the input is the *system message* that is prepended to all conversations to encourage special behavior. Such messages may vary depending on the application but could, for example, be: "You are a helpful AI assistant ...". Examples include giving instructive examples (few-shot learning) [28] or even adding an instruction such as "think step by step" (zero-shot learning) [44].

Maintaining a prompt over time can be tedious as they tend to be brittle and require adjustment for small changes in requirements. To this end, automatic prompt generation tools such as DSPy exist [45]. They circumvent the many pitfalls of optimizing the prompt by hand, in this case by stacking PyTorch-style prompt modules instead. While DSPy shows promise, it is still in early development and may still not be practical for that reason.

2.6.2 Hyperparameters for Generation

After applying an LLM to an input, a probability distribution for the next token is given over the vocabulary. The general purpose of language models is to construct the sequence with the highest probability. However, given the distribution of the next token, it is not obvious how to generate subsequent tokens which will give the highest probability to the overall sequence, as greedily choosing the most probable token is no guarantee. Instead, sampling from the distribution is often used, combined with hyperparameters which alters the distribution. Consequently, different values of the hyperparameters will elicit different behaviour from the model. The default setting for the model might not align with the best option for the task. Therefore, the performance can benefit from tuning the parameters, which includes:

- Temperature
- Top-k
- Top-p
- Local typicality
- Conditional probability cutoff

These can both be tuned in isolation and in combination with each other.

2.6.2.1 Temperature

The final softmax layer, which creates the distribution, is computed using the logits. The temperature simply divides the logit by the temperature value in the computation as:

$$\text{softmax}(z_i, T) := \frac{e^{\frac{z_i}{T}}}{\sum_j^K e^{\frac{z_j}{T}}} \tag{2.20}$$

When the temperature approaches 0, the probability of one token will approach 1. On the other hand, increasing the temperature will create a more uniform distribution. Thus, lowering the temperature often leads to higher quality generation but less diversity [46]. However, for the purpose of our project, a decrease in diversity is no concern, and we believe a low temperature will benefit the summaries.

2.6.2.2 Top-k

Top-k is an approach which truncates the distribution to a smaller set of tokens to sample from. This is done by simply choosing the tokens with k highest probability, re-normalizing the distribution and sampling the next token from the new distribution [47]. This way, the sampling is restricted to more probable tokens. Formally, the top-k vocabulary $V^k \subset V, |V^k| = k$ of the distribution $p(x_i|x_{<i})$ is found by solving the optimisation problem

$$p = \max_{V^k \in \mathcal{P}(V)} \sum_{x_i \in V^k} p(x_i|x_{<i}) \tag{2.21}$$

where $\mathcal{P}$ is the power set operator. The distribution is then re-scaled by

$$p'(x_i|x_{<i}) = \begin{cases} p(x_i|x_{<i})/p & \text{if } x_i \in V^k \\ 0 & \text{otherwise} \end{cases} \tag{2.22}$$

Choosing a suitable k for sampling is not trivial, and depends on the context. If the distribution is flat, choosing a small k risks creating bland text. Choosing a large k with a concentrated distribution will increase the probability of generating an improbable token. We believe the distribution will not be flat as we operate in a specific context. Hence, small k values will be favorable.

2.6.2.3 Top-p

Top-p or Nucleus sampling restricts the sampling to the smallest subset of tokens whose cumulative probability minimally exceed p [47]. The top-p vocabulary, $V^p \subset V$ is the smallest set such that

$$\sum_{x_i \in V^p} p(x_i|x_{<i}) \geq p \tag{2.23}$$

Then, the distribution is re-scaled equivalently as in top-k. Similarly to top-k, we believe a small p will be beneficial to truncate the distribution to the most probable tokens.

2.6.2.4 Local typicality

Local typicality is another parameter for truncating the distribution. It compares the similarity between the conditional probability of predicting a token and the expected conditional probability of a random token [48]. By defining a language process $\mathbf{X} = \{X_i\}_{i=1}^{\infty}$ and the entropy rate H of a stationary, ergodic language process as

$$H(\mathbf{X}) = \lim_{t \to \infty} \frac{1}{t} H(X_1, ..., X_i) \tag{2.24}$$

Then, locally typical sampling can be defined as sampling from the subset V^l where V^l is the solution to the subset optimization problem:

$$\min_{V^l \in \mathcal{P}(V)} \sum_{x_i \in V^l} |H(X_i|\mathbf{X}_{<i} = \mathbf{x}_{<i}) + \log p(x_i|x_{<i})| \tag{2.25}$$

$$\textbf{subject to} \sum_{x_i \in V^l} p(x_i|x_{<i}) \geq \tau \tag{2.26}$$

Similarly to top-p sampling, τ is a hyperparameter in the range of 1 to 0, and the distribution is also re-normalized. In practice, the methods restricts the sampling to tokens which have a negative log probability of a certain range from the conditional entropy. The range is determined by τ, which determines how large a probability mass from the original distribution should be included. Just as top-p, the truncated distribution will be constructed of the most probable tokens whenever the original distribution mass is concentrated to a few tokens. It is when the distribution turns flat where local typical sampling outperforms, and it is an especially promising prospect for abstractive summarization according to the original study [48].

2.6.2.5 Conditional probability cutoff

This parameter creates a truncated distribution by considering the probability of individual tokens rather than the collective. In short, if $p(x_i|x_{<i}) > \epsilon$ holds, then the token should not be truncated [49]. The recommended range for ϵ is from 3×10^{-4} to 9×10^{-4}. As probable tokens will be favoured for generation, we think a higher conditional probability cutoff will be beneficial.

2.6.3 Training

Out of the box, the pre-trained LLM will not have been sufficiently exposed to the Swedish medical data required for the task. There are currently two main methods to inject LLMs with proprietary data, Retrieval Augmented Generation (RAG) and additional training [50]. As RAG mainly involves information injection specific to single questions and not a broader task, this project is concerned with a training approach.

There are two main types of additional training, domain adaptation and fine-tuning. Domain adaptation is the practice of continuing the self-supervised learning of predicting the next token in a given corpus. This technique allows the model to become more familiar with tokens in the target application's domain and more accurately predict such tokens. This allows the model to learn more long-term concepts within a specific domain [50], which in this case could be medical terms or acronyms. Furthermore, domain adaptation could be applied to English-based models such as Llama-2 to improve performance on Swedish knowledge tasks. Fine-tuning, on the other hand, involves training a pre-trained model further on a task-specialized dataset of input and target output and is more tightly connected to the downstream task. The input data is constructed to mimic a question for the model, and the output is what the user would expect as an answer from the model. In the case of summarization, the set-up requires the text to summarize and a gold standard summary. Then,

next token prediction is only completed on the summary with the gold standard as correct class, ignoring the instruction.

2.6.3.1 Low rank adaptation

As suggested by the name, LLMs have numerous parameters, and hence, it can be expensive to train the model further. Practically, it requires hardware with sufficient memory to store gradients and updated model weights during backpropagation, and it takes considerable time. To bypass the problem, extensive research has been conducted in the field of parameter efficient finetuning (PEFT) [51] [52]. The idea is that rather than performing complete parameter updates, only certain or added parts of the model are updated during training. One such method is low-rank adaptation (LoRA).

LoRA freezes all of the original model weights and injects trainable rank decomposition matrices into specified layers of the transformer architecture [53]. The method can reduce the number of trainable parameters by up to 10000 while still maintaining performance from full parameter training. Furthermore, LoRA performs better than full parameter updates when the number of data points are less than a thousand.

LoRA makes use of a scaling factor γ to scale the parameter updates, similarly to a learning rate. The purpose of γ is to reduce the need to update hyperparameters concerning generation after training. In the original paper, γ depends on the rank r and a hyperparameter α according to $\gamma = \frac{\alpha}{r}$. For higher ranks, $r > 32$, gradients start to collapse, resulting in slower learning, and thus higher ranks perform equally to lower ranks [54]. The gradients stabilize when $\gamma = \frac{\alpha}{\sqrt{r}}$, which is referred to as rank-stabilized LoRA (rsLoRA).

2.6.3.2 Noisy Embeddings Fine Tune

Another technique to boost finetuning performance is adding noise to the embeddings during the training forward pass (NEFTune) [55]. Jain et al. report that using the technique boosts model winrates on the AlpacaEval benchmark across four different data sets for training. While the winrates improve up to 34.9 percentage units, ROUGE-L and BLEU scores are reduced, implying the method also decreases overfitting.

NEFTune works by adding a noise vector to the embedding vectors. The noise vector is constructed by sampling i.i.d uniform entries in the range $[-1, 1]$ and scaling the vector with $\frac{\alpha}{\sqrt{Ld}}$ where L is the sequence length, d embedding dimension and α is a tunable parameter, recommended between 5 and 15.

2.7 Ethical Considerations

Although LLMs have displayed excellent capabilities, their usage in the medical domain is questionable. The primary concern is the lack of explainability [56]. LLMs are without simple post-hoc explanations present in systems such as decision-trees

and linear regression. Instead, LLMs output are constructed from stacked transformer layers which in turn consist of stacks of multi-head attention, skip layers and non-linear transformations [56].

Chain of thought (CoT) prompting is a popular method to get a reason for the model output [57]. CoT prompting is a type of in context learning where examples are provided to elicit a step-by-step reasoning of the model output. An example is provided in the Appendix, Figure A.1. Unfortunately, it is not plausible in the situation of summarization as it would require large instructions to the model, bloating the context in the process.

Furthermore, explainability is not only an ethical concern but also a legal one due to the European Union (EU) AI act [58]. The EU AI act was accepted in December 2023 and imposes obligations on AI systems depending on their classification [59]. AI applications used in health are deemed high-risk, hence requiring transparency. The bill states:

"High-risk AI systems shall be designed and developed in such a way to ensure that their operation is sufficiently transparent to enable users to interpret the systems output and use it appropriately." [58].

The three main notions of model transparency are simulatability, decomposability and algorithmic transparency [60]. Simulatability refers to the concept that a model is transparent if a person can examine the complete model at once, for instance, if all computations can be performed by hand in a reasonable time. Decomposability in turn is the idea that each part of the model has an intuitive explanation. Finally, algorithmic transparency promises a unique solution of the training, even for unseen data. Although none of these notions holds true for an LLM, post-hoc methods concerning the explainability of a single generation exist, which could make the output sufficiently transparent to the user. Such methods include Sequential Integrated Gradients (SIG) and Shapley values to highlight the importance of each feature in the input in the model's generation [61] [62]. Adding such methods could ensure the system is legal to use in accordance with EU AI act, however it is out of scope for this project.

On the other hand, current practice at Swedish hospitals is to not employ AI models as clinical tools, but rather as aid which should not be relied upon completely. This is simply done with a disclaimer which circumvents current legal issues and the hospital staff is still completely responsible for the medical care. Nevertheless, ethical issues still remain.

2.7.1 Misguiding information

When generating text, the model can create misguided summaries both by hallucination and exclusion. Exclusion is not unique to LLMs as classic algorithms as TextRank may also exclude vital information. Hallucination on the other hand is a problem which does not exist in extractive summarization, as their output is entirely based on the given input. This could potentially be a reason enough not to employ an LLM in the given setting.

Regarding errors, Van Veen et al. study concludes that GPT-4 makes fewer factual errors than humans summarizing radiology reports, patient questions and progress notes [26]. It is not clear whether this applies in general, but it hints that systems more accurate than humans are possible to create, using LLMs. However, LLMs are still not without fault even if errors are reduced compared to the human gold standard, and the question then shifts to responsibility. Regarding general machine learning and the question of responsibility, Matthias argue it should not be considered on the individual level, but rather the collective of stakeholders involved [63]. Grote and Berens acknowledge the claims by Matthias but also note that there is still not sufficient legislation in the medical field, and in practice, individual responsibility still exists [64]. Although it is not clear who would be at fault, they are adamant that physicians can not be held responsible for the results of algorithms they do not understand. On the other hand, Durán and Jongsma argue AI systems are similar to other non-trivial technologies used by physicians, such as a magnetic resonance imaging (MRI) system. If a physician makes a poor decision based on a malfunctioning MRI, Durán and Jongsma claim the physician is still responsible [65]. It is clear arguments exist for both sides currently. EU has moved to fill the existing gap of responsibility in terms of consumer rights. The proposed AI Liability Directive (AILD), aims to make it easier to handle compensation claims for harms caused by AI-systems [66]. Unfortunately, in its current form, there exist liability gaps for AI-caused patient injuries which AILD does not cover [67]. However, the proposition itself proves that the extensive discussion of the topic is not in vain, and a conclusive solution could exist soon.

2.8 Related Work

Two previous works which are closely related to our project are presented below. Tang et al. make use of GPT-3.5 and ChatGPT to generate summaries of systematic reviews of medical evidence data [68]. In a zero-shot fashion, without extensive prompt engineering, they conclude the models produce comprehensive summaries more than 75% of the time, which they also identify has a strong correlation with high quality summaries. On the other end of the spectrum, missing important information is the primary reason human evaluators dislike summaries. Fabricated errors were deemed as the secondary reason as indicator of poor quality, and it was found the human reference summaries had a higher frequency of fabricated errors compared to the best performing model. Yet, human-produced summaries were still preferred by the evaluators. Apart from a difference in summarization material, the report differs from ours in that it makes use of a proprietary model.

Van Veen et al. improve the quality of summaries by applying quantized low-rank adaptation (QLoRA) to fine-tune eight different LLMs for four different summarization tasks: Radiology reports, patient questions, progress notes, and doctor-patient dialogue [26]. GPT-4 outperforms all other models in the study, however, all of the models gain improvements of summary capabilities from finetuning. Similarly to Tang et al., they find GPT-4 to produce fewer errors than humans, but also to be more complete, identifying conditions unnoticed by humans. Consequently, GPT-4

summaries were preferred 36% of the time compared to human at 19%, with a tie at 45%. Furthermore, the study suggests additional prompt engineering could further augment the benefits of finetuning. While this projects shares a lot in both approach and goals, it differs from ours in the group of models that were initially evaluated and that they used QLoRA instead of just LoRA. Most importantly, both of the presented works are English tasks whereas ours is Swedish.

2.9 Purpose

Due to the recent displays of prowess for LLMs of automatic text summarization in the medical domain, and the numerous techniques to alter their behaviour for down-stream tasks, we hypothesize modern LLMs can achieve human level results on the task of Swedish-medical summarization. Specifically, it should be possible to make use of an open-source LLM to acheive results not worse than human level, considering mentioned works primarily have focused on the best performing models out-of-the-box which currently happen to be proprietary. Stacking techniques like prompt engineering, supervised finetuning and domain adaption combined with a suitable model choice will make the model perform better. Despite the shortcomings of automatic evaluation strategies, they still provide sufficient correlation for human preference and are hence valuable as guidance during development to achieve human-level results. The possible pitfalls of using an LLM are convincing hallucinations and untraceable output.

3

Methods

The project explored the capability of LLM summarization in the medical domain. First, we created an evaluation procedure which uses a small synthetic dataset, produced by a clinician at Sahlgrenska. It automatically evaluates a model and scores its ability to produce accurate summaries, according to ROUGE-L, BLEU and METEOR. This procedure was then used to test a number of models, of which one was picked. The model was then domain adapted, finetuned using LoRA and finally hyperparameter tuned. Afterwards, we carried out an expert evaluation to compare the success of the final model to that of human made summaries. This was done by sending out a form to a number of clinicians which asked them to compare different summaries and score them along aspects like relevancy and hallucinations.

3.1 Data

The data used in the project consisted of a mix between real and synthetic data, depending on the optimization step in the pipeline. Data was synthetic for instances where real-world data was inaccessible for various reasons, such as privacy. Hence, all journals and other documents regarding individuals were synthetic, created by clinicians who were part of the project. The clinicians created the data from experience and did not use real data as a template. Real-world data includes study material compiled by medical students at Karolinska Institutet (KI) which is publicly available.

3.1.1 Summarization task

For the summarization task, the data consisted of the synthetic medical documents, used to prompt the LLM for a summary.

A synthetic example of a medical visit was provided by a specialist assistant nurse working with vascular surgery, shown in Figure 3.1. The text marked in yellow is extracted from the patient database ELVIS and printed to a physical paper, and the italicized text is what a nurse would then typically add by hand. The italicized information would then later be added to MELIOR, a journal system.

23-01-19	09:20	F	Bosse Bengtsson	19XXXXXX-XXXX	BENISCHK	ÖVRIGT
Pat fr avd XXX. Ringer och meddelar pat tid. Pat transp bokad fr avd XXX. Ej bokat åter då pat ev blir inlagd på 138.	*HT, DM, Op Fem-pop hö (-20),*			*Krit isch hö ben med osteitbild. Sår dig 1 hö fot*		*ABI FOTO*

Figure 3.1: A synthetic data example. Notes produced by the doctor use special conventions and acronyms to speed up the notetaking process.

The italicized text is divided into three categories, medical history, current symptoms/reasons for visit and next step care plan. To minimize the work to scan for such categories in all of the patients medical records, the model will do such for all medical records compiled during each day and return as a bulleted list. As a consequence, the input will be of medical records from a single day and output will be of a bulleted list for each category. This is the specific task which was optimized in the project. A short example (in Swedish) can be found in Table A.1 in the Appendix.

3.1.2 Training

The training data consisted of three datasets: domain adaption (DA), questions & answers (QA) and finetuning (FT). While DA and QA are in some senses both cases of domain adaption, their purposes differ.

DA contains samples of uninterrupted medical texts and serves to increase the models general exposure to Swedish and medical tokens. It is the largest overall dataset at around 8M tokens and was used with continued pre-training.

QA is around 1M total tokens of questions and answers pair. It was included since simple exposure to only DA-style data has been shown do degrade instruction comprehension whereas domain adaption through tasks does not [69]. To train the model on QA, supervised finetuning was used. An example of question and answer can be seen in Table 3.1.

Question	Hur sker inandning i vila?
Answer	Kontraktion av diafragma och externa intercostalmuskler ökar lungans volym - ökning i lungans volym leder till en trycksäkning och luft kan komma in i lungan - lungan är elastisk (utspänd)

Table 3.1: Question and Answer sample. An English translation can be found in the Appendix, Table A.2.

Concerning summarization finetuning, we used the synthetic data produced by the doctors as a baseline to generate more examples. The synthetic data is generated by taking the examples from the doctors and iterating through to create similar examples using GPT-4. To create more variation, GPT-4 was prompted to include an element from a list of 334 diseases and symptoms in the anamnesis. Subsequently,

the generated samples were examined to ensure they adhere to the template and the summaries do not include any false information. A complete overview of the set sizes can be found in Table 3.2.

Task	**Samples**		**Tokens**	
	Train	Validation	Train	Validation
Domain adaptation	2.1K	60	8.8M	245.8K (5%)
Questions & Answers	16.7K	878	1.1M	56.6K (5%)
Finetuning	217	12	218.8K	12.5K (5%)

Table 3.2: Number of samples and tokens in each data set

3.2 Evaluation

To quantify the success of the project, both automatic metrics and human based evaluation were used. We used metrics to evaluate performance during development and later human expert evaluation for the final evaluation.

In terms of metrics, regular n-gram based techniques were used, namely ROUGE, BLEU, and METEOR. The performance of this project was also compared to works by Tang et al. who use GPT-3.5 for medical summarization [68] and van Veen et al. who apply LoRA to finetune language models [26]. Furthermore, the metrics was also used to compare performance between different models in the project.

The evaluation procedure was created by first gathering a selection of description and summary pairs of fictive patients curated by a clinician, which in our case totaled to 24 test cases. The pseudocode is described below in Algorithm 1, where LLM is a function LLM : string $\rightarrow$ string and the operation $I + d$ describes the joining of an instruction with a particular description, even though in reality they are not simply appended. All metrics are functions which given a list of candidate-reference pairs will generate an aggregate value. Also, we define x as the reference summary, and $\hat{x}$ as the candidate summary for a given description d.

To assess the shortcomings of metrics, human evaluation was also required. Important aspects for human evaluation of a summary include relevance, coherence, redundancy, fluency, consistency and contradiction [21]. For the project at hand, the most important aspects were relevance, consistency and contradiction. It is crucial all important information from the source documents were included and does not contradict itself. Additionally, the chronological order of the source documents are critical to explain the patients journey. Fluency and coherence were of less significance as medical journals usually are low in lexical coherence [70].

While expert evaluation would have been interesting to do throughout the project, the time offered by the experts was limited, so only a final test was conducted. Moreover, the automatic evaluation procedure allowed to test even minimal changes to the setup at any point, which was practical.

Algorithm 1 Automatic Evaluation

Require: A list of description and reference summary pairs:
$P = [(d_1, x_1), \ldots, (d_n, x_n)]$
Require: An instruction prompt I
$L := []$
for $(d, x) \in P$ **do**
$\hat{x} \leftarrow \text{LLM}(I + d)$
append $(\hat{x}, x)$ to L
end for
$r \leftarrow ROUGE(L)$
$b \leftarrow BLEU(L)$
$m \leftarrow METEOR(L)$
return (r, b, m)

3.2.1 Form

The evaluation was conducted through an electronic form, sent to the participants. There were 31 volunteering physicians which the form was sent to. The form was divided into ten sections where each section concerned one collection of coherent notes and three summaries, human reference, our model and TextRank. The evaluators would read the notes and the presented summaries and score them on six questions, which can be found in Table 3.3. The question were constructed to cover the topics of relevance, consistency and contradiction of the content. Each evaluator would answer five sections, with the option to do five additional.

Question	Scale	Best value
How good do you think the summaries are overall?	1-10	10
How relevant did you think the content of the summaries was?	1-5	5
Does the summary contain any false information?	1-5	1
Is the summary missing any information that should have been included?	1-5	5
Would the summary have been useful in your work?	1-5	5
How well does the summary use medical terminology?	1-5	5

Table 3.3: The questions used in the evaluation form. Best value column indicates which value is the most favourable for a summary on the given question.

To prevent the evaluators from focusing on the structure, whereas the actual interest

of the form is to assess the content, we restructured the TextRank summaries to mimic the format of model and human summaries. We completed the reformatting by hand on the ten examples included in the evaluation form.

The results from the evaluation were used in a Bayesian repeated measures analysis of variance (ANOVA) to test whether there was a significant difference between the models. One such test was performed between each of the questions.

3.3 Model Selection

For this project, an *open source*, *pre-trained* LLM was used. Training a model from scratch is a serious effort, and in many cases not necessary due to the availability of high quality pre-trained models. One vital constraint the project had to respect is the privacy of patient data. While the project used synthetic data during development, future systems will in theory apply these techniques to data concerning real patients. It is therefore important these new systems are designed with the goal of data not leaving the hospital. This means that everything has to be able to be ran locally, without relying on external APIs that process the data, ruling out for example GPT-4 and Claude 3. Running the model on local hardware ensures that data does not leave the hospital, which is a major concern of the extended project at Sahlgrenska.

There exists numerous LLMs and it is pivotal to employ one which suits the task. A promising candidate is Llama-2 [27]. Besides scoring high on notable benchmarks, it comes in several parameter sizes (7B, 13B and 70B), meaning there is potential to scale up. Furthermore, Sahlgrenska has indicated a preference for Llama-2, for similar reasons, especially with a comparison between 7B and 70B.

While general models like Llama-2 show a lot of promise, there is also a need to consider more specialized models. One such possibility are the pair of open source medical models called MediTron-7B and MediTron-70B which are domain adaptations of Llama-2 on medical tasks [71]. Consequently, the models achieve impressive results on medical benchmarks.

Another challenge was that of the language barrier that exists between most LLMs and this new system. Most relevant models are designed with English in mind, and their performance on Swedish tasks may be subpar. To this end, the Swedish model GPT-SW3 was explored, which was specifically designed with Swedish in mind [72]. However, the performance of GPT-SW3 on Swedish tasks is not established due to the lack of suitable Swedish benchmarks.

Two additional models were also tested, Mixtral-8x7b and Falcons AI medical summarization. Mixtral was interesting due to being one of the first open-source mixture of experts (MoE) LLMs, with impressive benchmark results [73]. The selection of Falcons AI was justified through it being a dedicated sequence-to-sequence summarization model with an encoder, decoder architecture. Conclusively, the complete list was:

- Llama-2-(7b/13b/70b)

- MediTron-70B
- GPT-SW3
- Falcons AI - Medical Summarization
- Mixtral-8x7b

The performance of these models was evaluated by metrics described in section 3.2. The TextRank algorithm was also evaluated on the task as a baseline to compare against. Next, the best performing model was used for further optimization.

3.4 Model Tailoring

After choosing the model, the generation pipeline was optimized using the following techniques. Different combinations of the presented techniques were tested in order to find the best configuration.

3.4.1 Prompt Engineering

The type of prompts which was explored informs the model it is helping with a medical task where it is important to not make mistakes regarding patient information, and to clearly state when it does not know the meaning of the provided information, such as abbreviations. It was told to summarize the patients medical history with the provided medical notes. An example prompt would be:

```
You are a helpful medical assisstant, helping doctors
and nurses by summarizing information about patients.
You always answer in Swedish.
You always answer within five sentences.
Below a patients anamnesis is given, delimited by
tripple backticks.
```
```
{day_i notes}
```
```
Think step by step and construct a summary of the
medical notes.
```

Listing 3.1: Example prompt used during development.

To evaluate the output for prompts, Promptfoo[1] was used. Promptfoo is a framework to automatically test and compare output from different prompts and models. The tests used was to ensure output is in Swedish, does not contain redundant pleasantry phrases, no contradicting information and if the factuality of the output matches the reference summary. The final prompt was used in all model combinations.

[1] https://www.promptfoo.dev

3.4.2 Hyperparameters

We used random search to tune the hyperparameters for text generation. This process is in theory tedious due to the number of different combinations of discrete hyperparameters that are possible, as well as the need to sample continuous ones. However, restricting ourselves to the smaller set of described parameters in Section 2.6.2 made the search more feasible. The search was executed by sampling a set of parameters and then evaluate the performance of the configuration. The evaluation procedure from the model selection step was re-used for the purpose.

The random search was also divided into two passes. The first pass rotated between using top-k, top-p and local typicality, sampling the parameters uniformly in ranges of [2,50], [0,1] and [0,1] respectively. The temperature was chosen uniformly in the range [0,2] and was used in all rotations. Then, in the second pass, the best performing parameter was kept in the set and instead rotating on the other parameters, while also adding the conditional probability cutoff. The best performing parameter, along with the temperature were increased or decreased by ten percent in each try as well. This way, we could identify which parameter gave the largest boost in performance and subsequently add more to the set. In practice it was found that only a few parameters improved performance by deviating from the default setting.

Note that these parameters only effect sampling-based generation and greedy decoding was tested separately. Also, the number of beams where always set to one.

3.4.3 Training

The domain adaptation was performed before the finetuning. The primary reason is because domain adaptation on a instruction tuned model will detoriate its capabilities to follow instructions [69]. As finetuning is similar to the instruction tuning performed on models to get them to follow instructions in the first place, the finetuning step can re-introduce the performance loss form the previous step.

As the domain adaptation had significantly more data available than the finetuning, and it was not the final training step, full parameter updates were used and rsLoRA was only used during finetuning. All training steps used the Cross Entropy Loss function.

3.4.3.1 Domain adaptation

Training on the medical corpus with next token prediction, the data was split into equal chunks in the length of the model context window, with a stop token between documents. The last chunk, which was smaller than the context window was discarded to simplify the process by not requiring padding. To test the difference in domain knowledge before and after training, the perplexity was calculated on each chunk in the test set. The mean and variance of the chunk perplexities were subsequently computed and used as measures.

The training hyperparameters were all based on the original Llama-2 paper pretraining, although some had to be changed to accommodate the smaller training

data [27]. Hence, the AdamW optimizer is used with $\beta_1 = 0.9$, $\beta_2 = 0.95$ and $\epsilon = 10^{-5}$. The learning rate has a linear warm-up of 20 steps, peak of 3×10^{-4} and a cosine decay. A weight decay of 0.1 together with gradient clipping of 1.0 is also used with a batch size of 1. The set-up was run for a single epoch.

For the question answering, training was performed similarly to a finetuning step. The pairs where concatenated into a single instance with clear delimitation of instruction, question and answer. Each such instance was deemed a sequence, also requiring a start and stop token. Hence, the instance in Table 3.1 would be processed as:

```
<s> [INST] <<SYS>>
Du är en hjälpsam medicinsk assistent som hjälper läkare och
    sjuksköterskor genom att svara på frågor.
Svara på svenska.
<</SYS>>
Nedan ges en fråga eller ett medicinskt begrepp.
<fråga>
Hur sker inandning i vila?
</fråga>
Svara på frågan eller förklara begreppet.
[/INST]
Kontraktion av diafragma och externa intercostalmuskler ökar lun-
gans volym - ökning i lungans volym leder till en trycksäkning och
luft kan komma in i lungan - lungan är elastisk (utspänd) </s>
```

During training, the model would only compute loss on tokens succeeding the `[/INST]` token. The instruction and prompt was different from the summarization task. Instead, it told the model it would be given either a medical question to answer or concept to explain. As there is no way to guarantee the length of each pair, padding is required although the loss is zeroed out on the pad tokens.

The hyper parameters in this step were based on the instruction tuning performed in the Llama-2 paper as well. This changes the learning rate to start at 2×10^{-5} and decay with a cosine schedule. The batch size is increased to 2. The other parameters were unchanged.

3.4.3.2 Finetuning

The finetuning was completed using rsLoRA. Matrices were inserted into all query and value projection matrices in the transformers. The rank was set to 64, $\alpha = 16$, dropout probability was at 0.1 and no update of biases. Additionally, the learning rate was set to peak at 2×10^{-4} compared to the learning rate used for the question answering. Furthermore, NEFTune noise level of 5 was used.

3.4.3.3 Combinations

Due to the more complex nature of training, compared to the other techniques, different combinations of training were tried to find the mix which worked best. This means finetuning was applied on the base model, question and answer model

as well as a domain adapted question answer model. This was tested as the general nature of current LLMs is mostly sufficient for all purposes and do not require domain adaptation, finetuning usually suffices. However, as can be seen in the case of Meditron, it can also prove to be valuable [71]. Conclusively, this means three independent models where developed, with six training runs in total. A visual breakdown can be found in Table 3.4.

	Domain adaptation	Question Answering	Finetuning
Model 1	✓	✓	✓
Model 2	✗	✓	✓
Model 3	✗	✗	✓

Table 3.4: The three different model compositions trained in the project.

3.5 Limitations

One of the more time consuming and computationally intensive parts of our project was finetuning a model. It required significant planning in terms of determining the dataset to use and its composition. Even though multiple model compositions were tested, it would not have been feasible to implement across multiple base models. For this reason, this project did not compare several finetuned base models and instead applied finetuning to the model that was deemed most promising.

As mentioned in Section 1.5, the complete project at Sahlgrenska intends to implement a chatbot to aid the medical staff. This project focused only on the subtask of summarizing information which the chatbot may use later on. In short, the project did not focus on building an application but instead on how to create complete summarizations.

3.6 Computational Resources

As previously mentioned, certain sections of the project required the use of significant computational resources. Such resources include specialized graphical processing units (GPUs), designed with AI tasks in mind, which for example use specialized datatypes to speed up processing of tensors.

Running inference on large models and especially training are two primary examples where such resources are required. To this end, we were granted access to Sahlgrenskas AI platform "RunAI" which comes equipped with 16 Nvidia A100 GPUs with 80GBs of memory each. Of these GPUs, we had access to 8 of them. These resources were more than enough, since the largest models explored were 70B (FP32) parameters or around 300 GB of memory. This means they will fit on four GPUs for inference, and eight for training.

4

Results

The results of the optimization steps are presented in this section.

4.1 Model Choice

The evaluation scores for the different models can be found below, in Table 4.1.

Model	ROUGE-L	BLEU	METEOR
Open Source			
`meta-llama/Llama-2-13b-chat-hf`	**0.351**	**0.163**	**0.383**
`meta-llama/Llama-2-70b-chat-hf`	0.343	0.078	0.317
`malhajar/meditron-70b-chat`	0.138	0.032	0.122
`AI-Sweden-Models/gpt-sw3-20b-instruct`	0.301	0.102	0.291
`Falconsai/medical_summarization`	0.104	0.104	0.088
`mistralai/Mixtral-8x7B-Instruct-v0.1`	0.290	0.051	0.230
Proprietary			
`openai/gpt-4-0125-preview`	**0.515**	**0.266**	**0.488**
`openai/gpt-35-turbo-16k`	0.366	0.131	0.310
Rule Based			
`summa-textrank`	**0.171**	**0.058**	**0.130**

Table 4.1: Comparison of models. Higher is better across all metrics. The best score in each category is bold, $n = 24$. GPT-4 significantly outperforms current open-source models, that at best perform similarly to the level of GPT-3.5-turbo.

From the evaluation, we found Llama-2-13b to be the best open-source model across all metrics, even comparing favorably to Llama-2-70b. This result was surprising, and an effort to explore it more was made in Section 4.1.1.

In general, we found that LLMs outperformed the two other approaches, rule-based algorithm (summa-textrank) and dedicated summarization model (Falcons AI - Medical Summarization). The main issue with the non-LLMs is the inability to categorize and format the output. The medical professionals we spoke to during development indicated the need for categorizing information regarding medical history, reasons

for seeking care, and actions taken, as it would mimic their manual procedure in Figure 3.1. Preferably, the information should be presented as one bulleted list under each header.

TextRank only extracts relevant sentences and places them one after another to form a summary and would require an extra post-processing step to create summaries adhering to the template. This makes the comparison of TextRank to the other models somewhat unfair. However, the time required to develop the post-processing methods needed for fitting the extracted information into the header-list-format was not justified within the given time-frame of the project. Even so, it is not clear if it would even be possible to complete the categorization step without veering outside of rule-based techniques. Furthermore, this demonstrates the flexibility of LLMs compared to classic approaches to the problem.

Another notable result was the very low score of MediTron-70b. Since it is a finetune of Llama-2-70b, it would be reasonable to expect them to perform similarly. However, there are two main issues regarding Meditron, which became evident when using the model. First of all, the instruction-tuned models described in the paper were not released, only the models for open-ended generation [71]. Consequently, a community finetune had to be used which was not proven to achieve the same results as the official Meditron model.

The second problem ties in to the first, which is that the open-ended generation Meditron is not trained to generate an end of sequence token. The generated summaries always included a trail of unrelated text after the generated, useful response. Similarily to TextRank, it would be possible to create a post-processing step removing the unrelated text, but it was disregarded due to time constraints.

Finally, the latest releases of GPT-4 and GPT-3 available at the time of experimenting were also included. GPT-4 is currently generally accepted as one of the best general purpose LLMs[1] and outperformed all other models in the evaluation. It was included to illustrate that the gap between the best closed-source models and the best open-source models is still significant. We also included GPT-3 since we found it interesting to see the increment in performance compared to GPT-4.

4.1.1 Scaling Effects

Traditionally, increasing the number of parameters in a model will make it more expressive which generally yields improvements across metrics and benchmarks. This was, for example, observed with the release of Llama-2 [27]. It is therefore notable that the best model for this problem was Llama-2-13b, which performed on par with Llama-2-70b while also being more than twice as fast during inference on average, see Figure 4.1. For this reason, it was decided to include another metric, BERTScore, as a sanity check. Although Llama-2-70b had a slightly higher BERTScore, it ultimately showed more or less the same results as ROUGE-L/BLEU/METEOR, confirming that increasing the scale may only be important up to a point. Therefore,

[1] At the time of writing, GPT-4-1106-preview holds the #1 spot on the LMSYS Chatbot Arena [74].

we chose Llama-2-13b as our base model for further optimization.

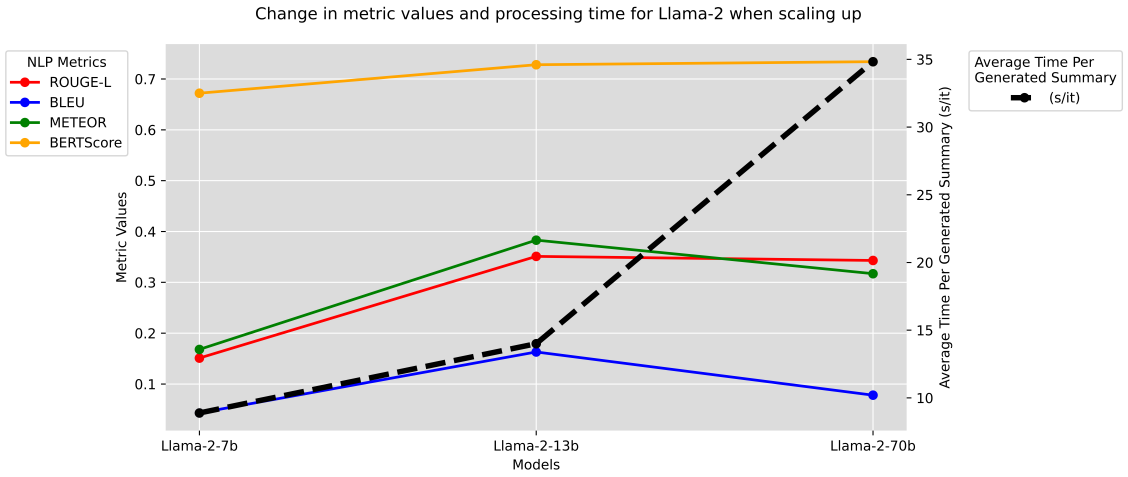

Figure 4.1: Comparison of how scaling up Llama-2 parameter count affects performance on NLP metrics and the average time spent per summary during generation. Performance on metrics shows diminishing returns when scaling up, whereas computation time grows drastically.

4.2 Prompt engineering

Given the selected model, Llama-2-13b, and application for summarization, a template for the prompt formed naturally. Llama-2 has a strict template that should be followed for performance, as it was used during the training. The template includes tags for a system message as well as the beginning and end of instruction. Additionally, the summaries were required to consist of three topics with bulleted information relevant to the topic at hand. These considerations resulted in the following prompt:

```
[INST] <<SYS>>
You are a helpful medical assistant who helps medical professionals by
    summarizing
information about patients.
Answer with a bulleted list for each category in the given template.
Answer in Swedish.
<</SYS>>
Below is the history of a patient during one day
<anamnesis>
*
</anamnesis>
You must select information that fits in the template below. Avoid
    unnecessary
information and only pick out material which relates to each heading. If
    relevant information is missing, leave the heading blank. Do not
    include any information in multiple headings.
<template>
```

```
*Sjukdomshistoria (The patient's diagnoses, medical history and risk
   factors
(e.g. diseases in the family))*

*Sökorsaker (Patient's symptoms and/or date of procedure)*

*Åtgärder (Planned examinations, treatments and measures)*

</template>
[/INST]
```

Listing 4.1: The final prompt. We found through experimentation that a mostly English prompt with specific template elements in Swedish gave the best results.

4.2.1 Recursive summarization

During development, we found that a subset of the collection of notes exceeded the context window of Llama-2 at 4096. To effectively summarize these collections as well, the text was split into even chunks that fit the context window with an overlap of 50 tokens between chunks. Then, all the chunks were summarized into plain text, and subsequently appended and used as basis for the final summary. This procedure was set up to be performed recursively until an intermediate text was small enough to fit the context window. In practice, it was never required to be performed more than once. The intermediate summaries required a different prompt, but less effort was spent on optimizing this prompt as it was used less frequently. The prompt can be found in the Appendix A.1.

4.3 Hyperparameter tuning

The most successful generation strategy was to use sampling rather than greedy decoding, and the most important change was to simply enable the renormalization of logits, which by default was not enabled. The full best configuration that was found is listed below in Table 4.2, compared to the original setting of Llama-2.

Hyperparameter	Tuned	Original
do_sample	True	True
renormalize_logits	True	False
typical_p	0.59	1
epsilon_cutoff	6.98e-04	1
temperature	1.67	0.6
top-p	1	0.9

Table 4.2: Table of hyperparameters for the generation configuration and their values. Floating point parameters were rounded down for space. A value of 1 means the truncation scheme is not used.

Also, it was a clear increase in performance gained by tuning the hyperparameters, which can be seen in Table 4.3.

Model	ROUGE-L	BLEU	METEOR
`Llama-2-13b-chat-hf`	0.351	0.163	0.383
`Llama-2-13b-chat-hf` + Best Gen-Config.	**0.392**	**0.195**	**0.414**

Table 4.3: Ablation study showing the performance increase of the generation config. Performance was improved across all metrics.

4.4 Domain adaption & Questions and Answers

After completing the continued pre-training on the DA dataset, the mean perplexity of the test chunks decreased from 6.2 to 5.1 whereas the variance shrunk from 3.3 to 0.7. The mean was reduced, but the variance was impacted the most, suggesting the model became more inclined to predict tokens in the test set, which consisted of medical texts.

The quality deterioration of instructions became evident as creating summaries with this model would start out well but decline into non-related output. The long texts in the training data is another reason for long outputs, as it resulted in long token sequences without a stop token, pushing 35 000 tokens. That is well above the context window of 4096 of the Llama-2 model. Hence, it essentially learns to generate a stop token less frequently than filling up the complete context window. This result further illustrates the issues mentioned regarding perplexity in Section 2.4.6.

The QA training resulted in the opposite containing mostly short texts. Somewhat mitigating the loss of instruction performance from the first step resulted in mostly short answers from instructions. The metric scores for both models can be found in Table 4.4. The model only trained with QA is presented as well, which performed better than both DA trained but worse than the base model.

Model	ROUGE-L	BLEU	METEOR
DA	0.03	0.01	0.05
DA + QA	0.10	0.03	0.07
QA	0.328	0.085	0.319

Table 4.4: Ablation study showing the performance decrease when domain adapting and question answers.

The loss of the DA training can be seen in Figure 4.2 and the loss of the QA training can be seen in Figure 4.3.

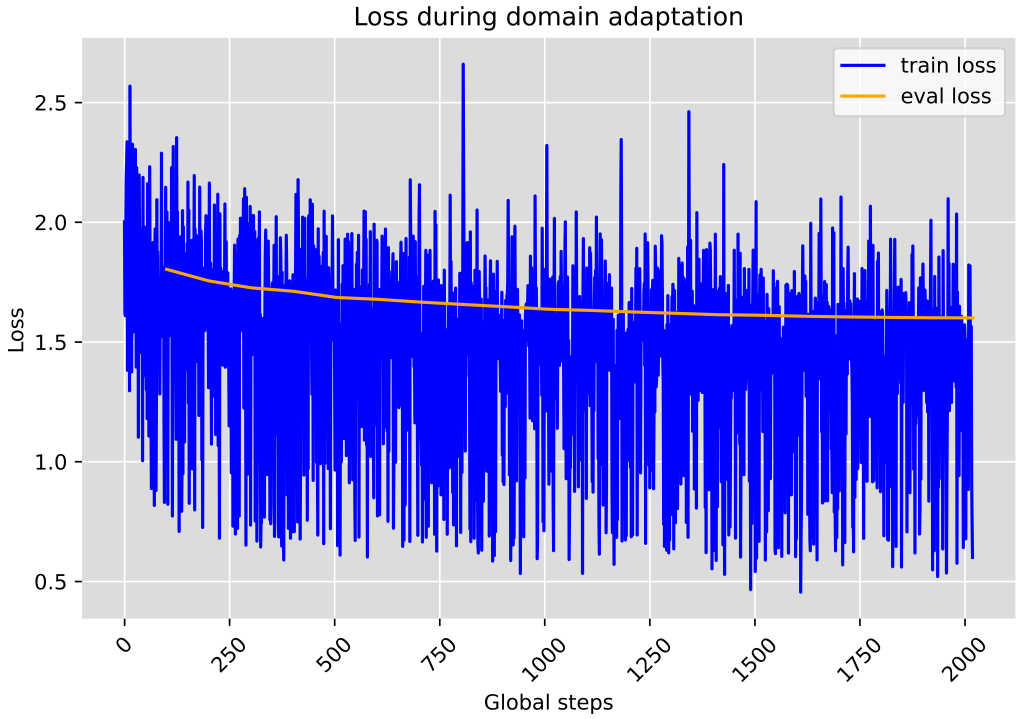

Figure 4.2: Train and evaluation loss during domain adaptation. Evaluation loss was computed every 100th step and can be seen to decrease.

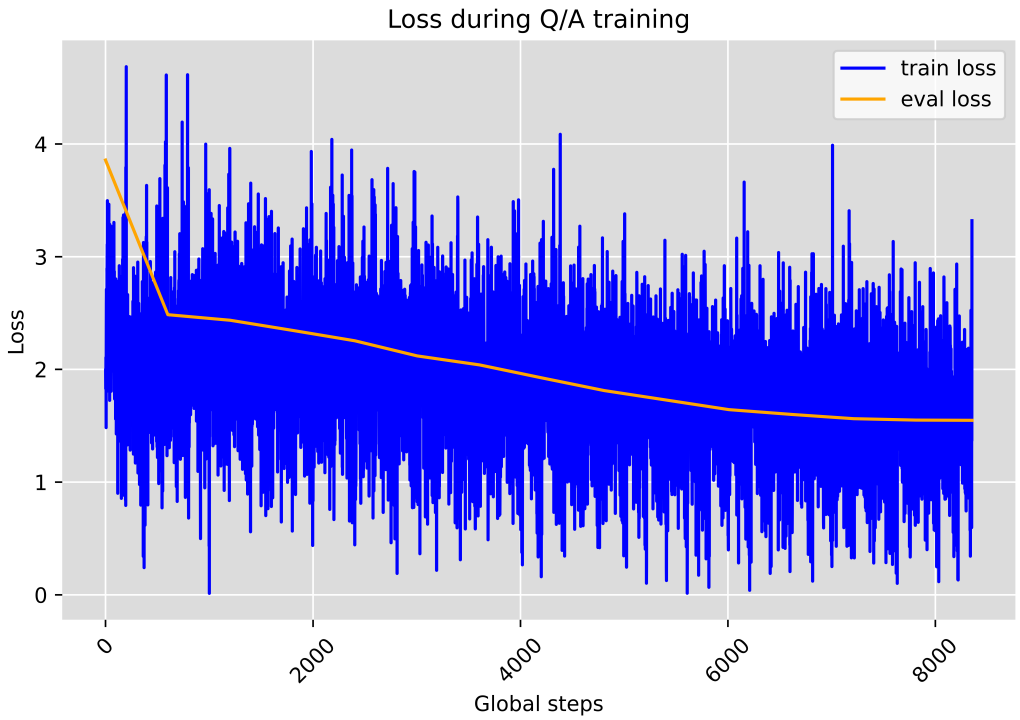

Figure 4.3: Training and evaluation loss of the task-based domain adaptation. Evaluation was performed every 600th step.

4.5 Fine-tuning

Given the rsLoRA configuration specified in 3.4.3.2, the total number of injected parameters reached 52 million (52,428,800), which constituted 0.4% of the total model parameters. Applying finetuning on the domain adapted models increases the performance again. The best of the three combinations in Table 3.4 was Model 3, which only used LoRA finetune. In an attempt to gauge the extent of instruction deterioration from the domain adaptation, the LoRA weights from Model 2 was applied to the Llama base model and evaluated as well.

Model	ROUGE-L	BLEU	METEOR
`Llama-2-13b-chat-hf`	0.351	0.163	0.383
Model 1	0.252	0.066	0.346
Model 2	0.347	0.112	0.349
Model 3	0.388	0.145	0.375
Model 2 LoRA loaded on base model	0.398	0.206	0.431
Model 2 LoRA loaded on base model + Best config	**0.435**	**0.225**	**0.466**

Table 4.5: Ablation study showing which model configuration performed the best. The last row displays the difference applying the best hyperparameter setting as well. The settings of hyperparameters can be found in Table 4.2 and the model configurations can be seen in Table 3.4. The performance of the Llama-2-13b base model is also included as comparison.

As can be seen from Table 4.5, this configuration created the best performing model, which was definitely surprising. It was expected the LoRA weights would perform best when applied to the model they were trained on. Substituting the base model creates a composition the LoRA weights are not optimized for. Yet, it once again reflect the decline of instruction capabilities seen from the domain adaptations which has been present across all results. Furthermore, it also conveys the difficulties in predicting the outcome of training, and the necessity to evaluate performance.

The scores of all model configurations can be found in the Appendix A.3. The training loss of fine-tuning on the QA model can be seen in Figure 4.4.

Comparing the result to other medical summarization studies, they seem reasonable, see Figure 4.5. Tang et al. do not perform any model optimisation beyond prompt engineering, thus the comparably weak results. Van Veen et al. on the other hand implement optimisation through QLoRA, temperature tuning and extensive prompt engineering. Furthermore, both the input and output of the other studies are shorter than the use of our model. The target summaries are specified as four and a few sentences respectively, which is comparably small to the summaries in our project and could also impact the outcome of the metrics. Although the summarization tasks might not be entirely similar, the biggest difference between the tasks lies in the language. However, as no other extensive studies on medical summarization in Swedish were found, these were the most suitable comparisons.

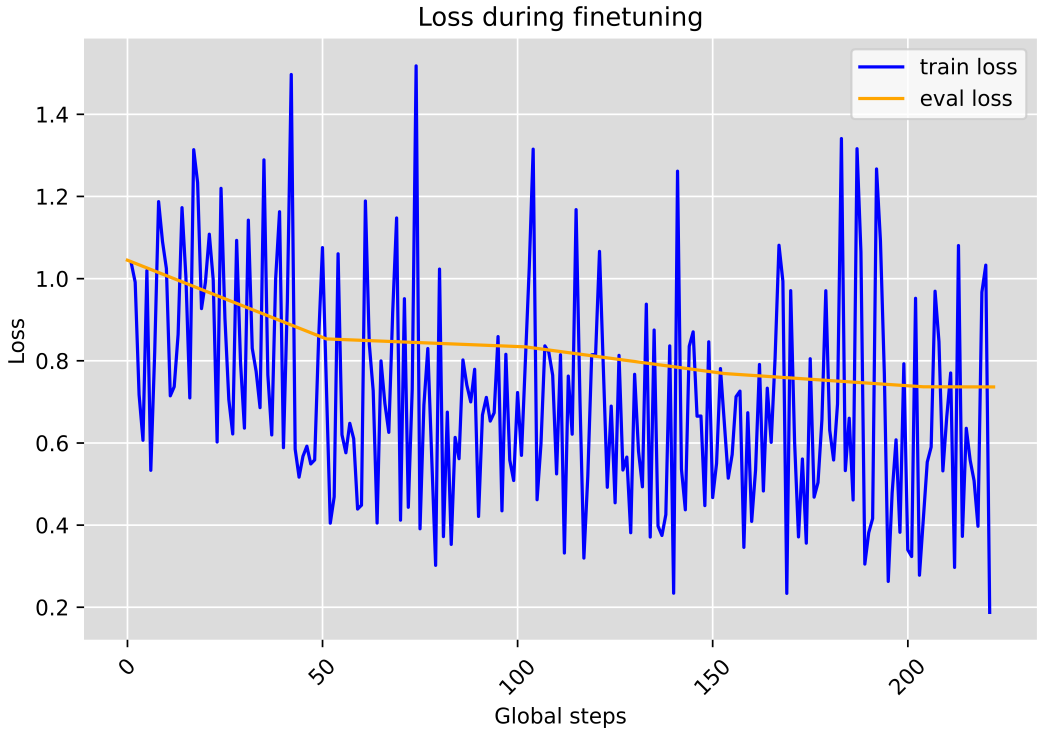

Figure 4.4: Loss during LoRA fine tuning for both training and evaluation. Evaluation was performed every 50th step and also once before and after complete training.

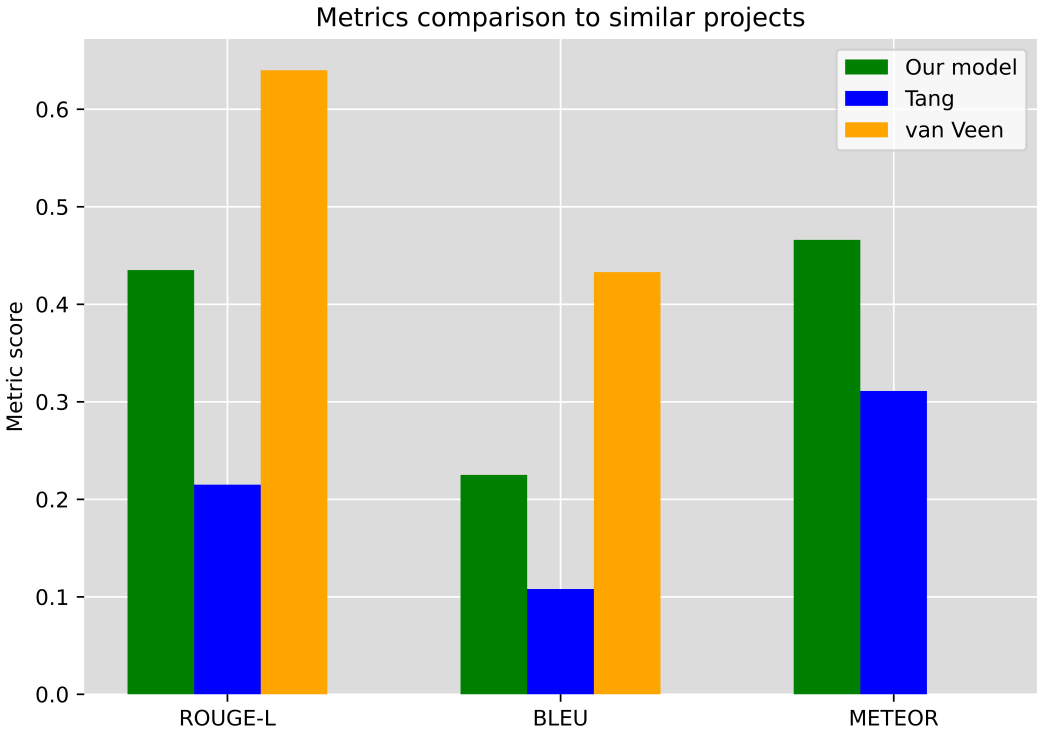

Figure 4.5: ROUGE-L, BLEU and METEOR score of the best performing models of two other projects focusing on medical summarization. Note the study by Van Veen et al., the METEOR score was not reported.

4.6 Expert Evaluation

Out of the 31 volunteers, only 7 answered the form. 6 of the evaluators completed 5 samples whereas the seventh completed 10, totalling to 40 samples. The results for the different evaluation questions can be found in the Tables 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11 below. Each table represents a Bayesian Repeated Measure ANOVA within a single question between all models, with estimates and 95% confidence interval of the mean. The standard deviation is also given. The model which we used was the best performing one found in Table 4.5.

Overall score				95% Credible Interval	
Variable	Level	Mean	SD	Lower	Upper
Intercept		3.406	0.223	2.948	3.841
Summarization models	TextRank	−1.139	0.210	−1.572	−0.723
	Ours	−0.537	0.208	−0.960	−0.138
	Human	1.676	0.215	1.231	2.104

Table 4.6: Mean of Overall scores from the evaluation.

Reapeated Measure ANOVA fits a linear model to the data according to $y = c+b*x$. In our case, y represents the estimated value for the question for a given model, c is the mean over all tested models, b is the regression coefficient for each model and x is a binary variable dependent on which model was used. Looking at the results from Table 4.6, the overall score average is 3.406 between all models. Then considering only the TextRank model, the score becomes $3.406 - 1.139 * 1 = 2.267$. The same method applies for the standard deviation.

Relevance score				95% Credible Interval	
Variable	Level	Mean	SD	Lower	Upper
Intercept		2.727	0.110	2.504	2.946
Summarization models	TextRank	−0.782	0.117	−1.022	−0.551
	Ours	−0.465	0.115	−0.707	−0.240
	Human	1.247	0.119	1.006	1.484

Table 4.7: Mean of Relevance scores from the evaluation.

False Information				95% Credible Interval	
Variable	Level	Mean	SD	Lower	Upper
Intercept		1.450	0.084	1.279	1.615
Summarization models	TextRank	−0.114	0.081	−0.280	0.045
	Ours	0.416	0.085	0.244	0.584
	Human	−0.301	0.082	−0.468	−0.142

Table 4.8: Mean scores of False information present in the summaries.

Missed information				95% Credible Interval	
Variable	Level	Mean	SD	Lower	Upper
Intercept		2.750	0.132	2.477	3.010
Summarization models	TextRank	−0.733	0.130	−1.003	−0.480
	Ours	−0.684	0.131	−0.949	−0.427
	Human	1.416	0.132	1.150	1.674

Table 4.9: Mean scores of Missed information from the documents.

Usefulness				95% Credible Interval	
Variable	Level	Mean	SD	Lower	Upper
Intercept		2.383	0.112	2.153	2.607
Summarization models	TextRank	−0.788	0.125	−1.042	−0.539
	Ours	−0.470	0.125	−0.732	−0.223
	Human	1.258	0.126	0.995	1.507

Table 4.10: Mean scores of the summaries Usefulness.

Medical Conventions				95% Credible Interval	
Variable	Level	Mean	SD	Lower	Upper
Intercept		3.592	0.144	3.297	3.873
Summarization models	TextRank	−0.227	0.103	−0.442	−0.022
	Ours	−0.418	0.104	−0.633	−0.209
	Human	0.645	0.107	0.429	0.853

Table 4.11: Mean scores of the models' proficiency in handling Medical Conventions.

The general trend from the results is that the human significantly outperforms both our model and TextRank. Our model tends to perform better than TextRank, but not significantly, as their lower and upper bounds are not entirely separated, which holds true for all topics except false information. For the topic of false information, our model clearly scores higher, albeit with a low mean of 1.87 out of 5. Nevertheless, it confirms the suspicion that the LLM solution is prone to hallucination.

Following, two examples are shown, one where our model is rated poorly and one of high quality. Also, the corresponding human summaries for the same instances are also given, which where evaluated as great for both instances. Figure 4.6 shows the example where our model was rated poorly and Figure 4.7 shows the example where our model performed well.

Our Model
Sjukdomshistoria (Patientens diagnoser, sjukdomshistorik och riskfaktorer (t.ex. sjukdomar i familjen))
 * Diagnoser: Hyperlipidemi, hypertension, förvirring
 * Sjukdomshistorik: Kroniskt subduralhematom på höger sida, hypodenst område
 * Riskfaktorer: Ålder (63 år), rökning

Sökorsaker (Patientens symtom och/eller datum för ingrepp)
 * Symtom: Våldsam huvudvärk, nedsatt kraft i vänster arm och ben, tilltagande huvudvärk, förvirring, lätt facialispares
 * Datum för ingrepp: 24-01-23 kl 1532

Åtgärder (Planerade undersökningar, behandlingar och åtgärder)
 * Planerade undersökningar: CT hjärna, ny CT hjärna vid klinisk försämring
 * Behandlingar: Observation, RLS-kontroll varannan timme, blodprov med Hb, infektionsvärden, elstatus, PK och APTT
 * Åtgärder: Operation/utrymning med planerad platsbrist, försök att överflytta under morgondagen

Human
Sjukdomshistoria (Patientens diagnoser, sjukdomshistorik och riskfaktorer (t.ex. sjukdomar i familjen)):
- 63 år
- Man
- Hyperlipidemi
- Hypertension
- Rökare
- Patienten står ej på blodförtunnande.

Sökorsaker (patientens symtom och/eller datum för ingrepp):
- Falltrauma 23-12-24
- Akutmottagningen SKaS: 24-01-23
- Nedsatt kraft vänster sida. Tilltagande huvudvärk. Förvirrad, Vaken (RLS 1). Ej orienterad i tid och rum. Något sluddrig i talet. Lätt facialispares. Uppegående. Haltar vänster. Dålig balans
- CT: Hypodenst område. Kroniskt subduralhematom (höger). Komprimerad lateralventrikel. Begynnande överskjutning av medellinjen

Åtgärder (planerade undersökningar, behandlingar och åtgärder)
- Operation/utrymning
- Blodprov med Hb, CRP, LPK, elstatus, PK och APTT innan överflytt
- Överflytt 24-01-24
- Inläggning för observation
- RLS kontroller varannan timme
- CT hjärna vid klinisk försämring och kontakt NK jour
- Fastande från midnatt
- Hibiscrub x 2

Figure 4.6: Comparison of summary between our model and the human. For this instance our model was rated poorly and the human summary was rated well, 1 and 7 respectively in overall score.

Our Model
Sjukdomshistoria (Patientens diagnoser, sjukdomshistorik och riskfaktorer (t.ex. sjukdomar i familjen))
* Diagnos: Wake up stroke
* Sjukdomshistorik: Patienten har känd HT, FF, och Dm typ 2
* Riskfaktorer: Patienten har haft förmaksflimmer

Sökorsaker (Patientens symtom och/eller datum för ingrepp)
* Symtom: Förlamning i höger arm och ben, svårt att prata
* Datum för ingrepp: 231113, CT hjärna med angio och perfuison, trombektomi

Åtgärder (Planerade undersökningar, behandlingar och åtgärder)
* Planerad utredning: CT hjärna kontroll utanför fönster för trombolys
* Planerad behandling: Thrombektomi och uppföljning via SKAS
* Nytt läkemedel: Tabl Atorvastatin 80 mg 1 × 1
* Byte av läkemedel: Från Eliquis till Pradaxa 150 mg 1 × 1
* Restriktioner: Under begränsad period olämplig att köra körkortspliktigt fordon
* Planering: Ingen planerad uppföljning via Sahlgrenska, överflyttning till strokeavdelningen SKaS

Human
Sjukdomshistoria (Patientens diagnoser, sjukdomshistorik och riskfaktorer (t.ex. sjukdomar i familjen)):
• I633 Cerebral infarkt orsakad av trombos i cerebrala artärer → AAL15 Intrakraniell endovaskulär trombektomi
• I109 Essentiell hypertoni (högt blodtryck utan känd orsak)
• I489 Förmaks flimmer och förmaksfladder, ospecificerat
• E119 Diabetes Mellitus typ 2-Utan komplikationer

Sökorsaker (patientens symtom och/eller datum för ingrepp):
• Stroke 23-11-13. Expressiv Afasi. Bortfall hö. Facialis pares
• NIHSS 23
• BT 190 systole (övrigt UA)
• CT: M1 ocklusion vänster med penumbra ca 50%
• Ej kandidat för Trombolys (blodförtunnande och Wake-Up Stroke)
• Trombektomi planeras

Åtgärder (planerade undersökningar, behandlingar och åtgärder)
• Trombektomi 23-11-13
• Strokevård enl rutin
• Trombyl t.o.m 23-11-16
• Tillägg: Tabl Atorvastatin 80 mg 1 × 1
• Byte: Eliquis → Pradaxa 150 mg x 1 (fr.o.m. 23-11-16)
• Uppföljning VGR SKAS Strokeavd
• Ingen uppföljning SU/S
• Restriktion: olämplig att köra körkortspliktigt fordon (3 mån)

Figure 4.7: Comparison of summary between our model and the human. For this instance both our model and the human summary was rated to be of high quality, 7 and 9 respectively in overall score respectively.

4.7 Compression

The major focus of this project was summarization of patient details. A challenging aspect of this was on the one hand to compress the information in a way that is both concise and easy to use for medical professionals while also remaining accurate. It is important that information is not oversimplified in order to be short. Comparison in length between our model, human reference, and original input can be found in Table 4.12.

On average, our model creates summaries which are about half the size of the input, which is slightly longer than the reference summaries. Note that due to the

	Mean	Standard deviation	Min	Max
Original input	740	619	56	2843
Reference	328	203	104	941
Model	393	134	215	789

Table 4.12: Comparison of token lengths of the original documents, reference summaries and model output.

structured format of the summaries, they contain a higher portion of newline tokens compared to the input. The model does have a lower standard deviation of length compared to the references, and overall a shorter range from minimum to maximum length as well. However, neither our model nor the gold standard managed to produce a summary shorter than the smallest input text. Overall, both approaches manage to reduce the amount of text. However, as seen in Section 4.6, the human references contain more relevant information and are generally better at compressing without oversimplifying the content.

5

Discussion

The following section discusses some key findings from our results and justification of methods.

5.1 Open Source vs Proprietary

There exists a large performance gap between open source and proprietary models. OpenAIs GPT-4 (and recently also Anthropics Claude 3 Opus) has distinguished itself as a highly capable model, significantly more so than the current best open-source offerings.

While using an open-source LLM was the goal of this project, it is still informative to compare it to the best proprietary LLM. As mentioned previously, open-source offers a higher degree of control, as well as, for example, a full overview of the architecture and the weights of the model. Even in the case of open source LLMs however, not all details are available. Publishing the exact details of the training data is still uncommon, meaning reproducing the results is impossible. Moreover, it presents issues when continuing the training phase. Naively continuing training with a new dataset and task often causes performance on the old task to drop, a phenomenon known as "catastrophic forgetting" [75]. As shown by Kirkpatrick et al., catastrophic forgetting can, for example, be mitigated by singling out weights that do not impact the new task and constraining how much they can change from their old value. While their approach only requires the original weights and not the accompanying dataset, this technique is computationally expensive in our setting [76]. A more efficient strategy was explored by Gupta et al. using LLMs, which incrementally increases the learning rate when training on the new dataset (called re-warming) [77]. They found that although re-warming worked in experiments, balancing the performance of old and new tasks was challenging.

Ultimately, working around catastrophic forgetting is a hard problem without access to the pre-training dataset. Open-source in traditional software allows for free access to the source code and the possibility to modify and compile yourself [78]. For that reason, it would perhaps be more accurate to call open-source models open-weight, as it is not possible to construct the weights without the data.

In an effort to create truly open-source models, Groeneveld et al. released the language model OLMo, together with all the code and data used during development

[79]. Their 7B variant performs comparably to the Llama-2 counterpart, which is promising but ultimately worse than the best proprietary models.

As evident by our evaluation, even our best setup scores lower than the best general purpose proprietary LLM on the problem of patient summarization, and it is unlikely to change with future models. High-quality open source models may become less and less prevalent, as the business of training LLMs may be relegated to only the few large corporations that can provide the computational capabilities. Future projects involving LLMs will, therefore, need to ask whether the sacrifice in performance is worth the level of control gained and if that level of control can itself be used to make up for the lost performance, via e.g., domain adaption and finetuning.

5.2 Parameters vs training

As seen during the model selection, the model with the most parameters does not necessarily have the best performance on the task. The following section will discuss other factors which can impact the performance of a model, with a focus on training data size.

Llama-2 was trained on 2 trillion mostly English tokens, of which approximately 3 billion tokens were Swedish [27]. GPT-SW3-20b, on the other hand, was trained on 300 billion total tokens, where 78 billion were Swedish [72]. Additionally, both Danish and Norwegian made up significant parts of the pre-training dataset, which are more similar to Swedish than English. Yet, Llama-2-13b still outperforms it on every metric in our study.

Kaplan et al. proposes that to avoid overfitting when training to a training loss threshold of 0.02 convergence, it is required that

$$D \gtrsim (5 \times 10^3) N^{0.74} \tag{5.1}$$

where D is the token size of the data set and N represents the number of parameters in the model [80]. This relation holds for neither Llama-2-13b nor GPT-SW3-20b which would require at least 33 trillion and 46 trillion tokens respectively. However, the Llama model is substantially closer to the target than GPT-SW3, suggesting it is better at generalizing tasks. Hoffman et al. give a less restrictive scaling of token size compared to parameters. Instead, they claim 205B as the optimal token size for a model of size 10B [81]. Linearly approximating their scaling law, a model of size 20B would require 430B tokens in the training set. For this instance, Llama-2-13b has a sufficiently large data set, indeed, it exceeds the optimal size. GPT-SW3, on the other hand, still does not have enough tokens.

In the technical report, GPT-SW3 reports similar benchmark scores to GPT-3 [72]. The two out of three tasks which GPT-SW3 reports better scores are question answering tasks whereas the third is a language inference task. This suggests GPT-SW3 might be over-fitted to tasks of answering questions rather than instructions,

which also is a substantial part of their instruction tuning set. Hence, the subpar results from GPT-SW3 could be a result of insufficient training data. There are certainly other factors that affect the training outcome beyond model and dataset size. The architecture of the model, the composition of the training data, the general quality of the training data, and the hyperparameters used for training are also important factors. In the case of GPT-SW3 and Swedish data, the number of inflections in the Swedish language can result in a drastically different tokenizer than an English-based model. Reasoning about what makes the models perform well is, therefore, complicated. It is also why it is important to create tests that are close to the actual use case and not simply go by standard benchmarks.

The composition of the training data could be especially important. Sorscher et al. observes how quality data pruning can beat power scaling laws and improve transfer learning [82]. On the other hand, Meta AI continues the trend by claiming they see continued improvements for Llama-3-8b training beyond 200B tokens, extending to 15T tokens [83]. In general, it performs better than the previous generation Llama-2-70b[1], with only a little more than a tenth of the number of parameters.

Disregarding the fact that our LoRA finetune was the training which mimicked the downstream task the most, it also had the most training tokens in comparison to the number of trained parameters. Obviously, neither had a number of tokens sufficient for the scaling laws proposed by either Kaplan et al. or Hoffman et al. The Meditron model we explored at the beginning of the project used 46.7B tokens during continued pretraining. However, this was an English domain adaptation and thus easier to find quality public data, whereas we were interested in creating a Swedish adaptation. Most existing Swedish public datasets are not focused on medical tasks. The ones we found were either NER or machine translations of poor quality. Both AI Sweden and Kungliga biblioteket (National Library of Sweden, KB) mention the difficulties of finding high-quality data in Swedish and the need to construct their own datasets [84] [85]. As neither have decided to release their datasets, a chicken race exists where no one wants to give up their advantage, but still neither has sufficient to create a model which substantially outperforms English base models on Swedish tasks. Considering the small reach of the Swedish language, we believe more collaboration is required in order to make significant strides within the Swedish NLP community. Alternatively, the transfer capabilities of modern models might be sufficient to make Swedish adaptations obsolete. However, that still requires Swedish training data to a certain degree.

5.3 Generated Data Commentary

Using LLMs to generate synthetic training data may be unwise since it could propagate potential biases and stereotypes present in the generating model. GPT-4, which was used to generate data for this project, has been shown to display certain biases that makes it potentially unsuitable for a medical setting, which, for example,

[1]At the time of writing, Llama-3-8b has a higher ranking than Llama-2-70b LMSYS Chatbot Arena [74]

was explored in a study by Zack et al. [86]. An attempt to mitigate this was made, using sampling from a list of 300+ diseases/conditions during generation, to increase diversity in patient generation.

The generated data could also include misinformation, e.g. including a fact in the summary which is not present in the description. While this can be a problem with real data as well, cases of misinformation generated by LLMs are often harder to detect than misinformation generated by humans [87]. This is even true when the detector is another LLM instance. This presents a challenge especially when cleaning the data.

Using a language model that employed language model-generated text for training is also not recommended. Shumailov et al. reports how generating data and then using it for training another language model, through multiple cycles, creates a distribution centered around a small number of tokens, reducing the probability mass of the tail [88]. This is undesired as low probability tokens are essential to the models, as they are often relevant to marginalised groups. Considering content on social media, which has historically been a source of training text for language models, has shifted from purely human-produced to mostly AI-generated, developers utilizing new content from them as training will make use of substantial model-generated data [89]. As the newly trained models will then be used to generate new content for social media, the spiral is started. Therefore, Shumailov et al. argue that efforts need to be made to ensure internet-crawled training data does not include content generated by LLMs [88]. As the only potential use of our model is within Sahlgrenska, and not to generate new data, we do not think it will be an issue to have used generated data.

As our task was fairly narrow, we used knowledge distillation (KD) in a fairly simple fashion, with GPT-4 as the teacher and Llama-2-13b as the student. Other research, however, has employed more involved setups, which also could have proven interesting for this project. For example Orca, a 13-billion parameter LLM that uses KD from GPT-4 demonstrates the virtues of including an explanation in the answer to the task, and emphasises task diversity during distillation [90]. Another more recent result is from Chen et al., who go beyond prompting to include an intermediary proxy model between the teacher and student model [91]. This setup uses a prior estimation directly from the teacher and compares it to a posterior estimation generated via a proxy model.

For this project, using data from real patients was never considered for ethical and legal reasons. Equally, the availability of human annotators for synthetic data was scarce and moreover, there was some disagreement among annotators on format and style. Using generated data was, therefore, the only option to get the amount of data necessary for using LoRA. In the end, the generated data helped to improve the performance on the task. Recent research by Wei et al., however, points out some virtues of using LLMs for data annotation [92]. On the whole, LLMs were both more factful and cheaper compared to human annotators.

5.4 Optimization Outcome

The training composition that achieved the best results was using the LoRA weights from the QA model (Model 2) applied to the base model. This was surprising as it seems most reasonable that the LoRA weights would perform the best on the model they were trained on. Furthermore, a LoRA finetune on a model not adapted would intuitively seem like a good candidate as well. However, both those compositions were outperformed. We can only hypothesize as to why that may be, as no straightforward logical explanation exists. The most sensible idea is that the LoRA weights incorporate the domain knowledge from the domain-adapted model, yet the Llama-2-13b base is still better at instructions. The original LoRA paper even states the low-rank adaptation matrices amplify some features in the original attention matrix [53]. In addition, they claim these features are related to the important features of the specific downstream task. These important features could be linked to the domain adaptation. Conclusively, this combination then mimics the complete process we would like to have performed, a domain-adapted model without the loss of instructions.

The practice of applying trained LoRA weights to different base models has not been established in the NLP community. Rather, the interest is on applying multiple LoRA weights to the same base model to achieve MoE behavior between downstream tasks [93]. This is reasonable as LoRA is lightweight to train, and there usually is no need to transfer weights from one base model to another. It is simply easier to just re-train on the new model. Nevertheless, for image generation through stable diffusion, this has been explored to a limited extent, with claims that it improves generation.

Another result that stood out was the result of the hyperparameter tuning. As mentioned in Section 2.6.2.1, we assumed a small temperature, at least less than one, would be suitable to control the output. Instead, we found our final configuration with a temperature of 1.67. We believe this is a result of the combination with the local typicality truncation scheme.

The temperature is applied before truncating the distribution, meaning truncation will be performed on the temperature-altered distribution. With the temperature set above 1, the altered distribution will be more flattened than without temperature. Consequently, the conditional entropy will be increased, and the topmost probable tokens will have a conditional probability too far from a random token. Hence, we believe the set-up becomes a shifted top-p where the absolute top tokens are not included. Why this gives better metric scores could be explained by the properties of human communication. To form effective communication, it is desired to minimize misunderstandings while still transmitting important information. This gives rise to a relationship between redundancy and duration of speech. Robust communication spreads out information evenly [94]. If this theory is correct, the human-crafted gold standard summaries should follow this pattern as well. Hence, when leaving out the most and least probable tokens from sampling, the generation mimics the concept of spreading out the information across the summaries. Thus, they should compare more favorably with the metrics. Furthermore, the model still manages always

to output summaries adhering to the given template, meaning that sometimes the temperature altering does not necessarily manage to increase the conditional entropy enough to leave them out, which works to our advantage.

Another possibility for the high temperature is the usage of language model generated training data. As discussed in 5.3, the procedure results in a model which next token distributions are more concentrated around a few tokens. The high temperature helps to reduce this effect by giving more probability mass to the tail of the distribution. To the contrary, the high temperature was observed as beneficial before the model was trained on the generated data.

5.4.1 Optimization target

When evaluating whether a new hyperparameter configuration or trained model instance performed better or worse than another, they where evaluated using the metrics. However, as mentioned in Section 2.4.6, these might not correlate well with human preferences. Hence, BERTScore was also used as an additional measure. Table A.4 in the Appendix shows the difference in performance between the best and worst hyperparameter setting, in terms of the other metrics, with BERTScore included. From this simple sample, it can be seen that the other metrics correlate decently with BERTScore, which in turn correlates more with human judgment.

On the other hand, this does not always hold true. During model selection, it could be seen that Llama-2-70b had a slightly higher BERTScore than Llama-2-13b while having worse score on the other metrics. The 13b version was still preferred as the difference was almost negligible. Also, GPT-4 had the highest score on all of the metrics, and as it has been seen to produce summaries of high quality, it is reasonable to optimize to the target.

All metrics could have benefited from more than a single reference summary per patient description. For example, using the mean or max over a list of references would have been a fairer judgment than using just one as we currently do. This would have especially been important in mitigating the problematic cases of ROUGE since more important n-grams would have been more frequent.

Even though the metrics were justified as basis of optimization, it is possible the human provided summaries where not of high enough quality to be considered gold standard. The mean of the scores given to the human summaries can be found in Table 5.1 below.

Most questions are within a reasonable range from the best possible value, except for the overall score. There seems to be a disconnect, beyond the different scales, between the overall score and the other factors. Therefore, it is possible another factor exists which impacts the overall judgement, one which we did not probe. Aspects mentioned in Section 3.2 which we did not inquire were fluency and coherence of the summaries. They were left out as they were not important to our project. However, the evaluators may still believe so, which could explain the disconnect. But, if there were other factors which contributed to the low overall scores, factors related to

Question	Mean score	Best value
How good do you think the summary is overall?	5.08	10
How relevant did you think the content of the summary was?	3.97	5
Does the summary contain any false information?	1.15	1
Is the summary missing any information that should have been included?	4.17	5
Would the summary have been useful in your work?	3.64	5
How well does the summary use medical terminology?	4.24	5

Table 5.1: Mean scores of human summaries in evaluation

the content of the summaries, then we might have optimized towards substandard summaries.

5.5 Performance compared to human references and TextRank

We hypothesized it was possible to achieve human-level performance on the task of Swedish medical summarization at Sahlgrenska. The results seen from experiments point to not yet.

Overall, the model summaries are slightly longer than the reference summaries, which could indicate they would, on average, contain more information. However, when also considering the standard deviation, minimum and maximum length, it is evident the human is more flexible, adapting the summary length to the input. This was also evident from the evaluation results, the relevance of information is ranked higher for the human reference compared to model output. Anyhow, neither method produces a summary shorter than the smallest input. This indicates the summaries might not strictly be compressing information and keeping the essence of information but rather focused on structuring the essence of information. Hence, short input documents might not be reduced in size but merely re-structured for readability through topic separation.

The model did outperform the TextRank algorithm on the topic of relevance, albeit not significantly. In fact, it is the topic which has the least overlap of intervals except for overall score. As explained in Section 2.5, the sentences compiled for summary, with TextRank, are ranked based on the similarity of words in common. This means

words that cover different subjects will not vote for each other and subsequently receive a lower vertex score. But, in the case of this task, it is important all subjects are covered in the output. This could explain TextRank's poor results. It would be interesting to explore whether the vertices with the lowest scores rather than highest scores would be more suitable as they might cover a wider range of subjects for this task. However, comparing the vertex scores to information gain (a high score is a low information gain) as discussed in Section 5.4, it is not trivial which sentences to choose to compose a summary that matches a human extractive summarization.

In terms of missing information, our model and TextRank performed equally poorly. The reason for our model's poor performance, we believe, is due to insufficient high-quality training data. This type of summarization task seems to still be difficult for LLMs, as other studied works usually has a smaller input and more coherent text as output [68] [26]. Thus, more task adaptation is required for the model to behave as desired. Although the evaluation loss (in Figure 4.4) seemed to converge with a few number of training steps, we think the generated samples could be of higher quality. They were cleaned for hallucination, but no effort was made to ensure all relevant information was included. Also, considering the general preference of the human summaries in the evaluation, more training data to mimic such behavior would improve the performance of our model.

The aspect in which our model distinguished itself the most was false information. Even though the overall mean was quite low, 1.87 out of 5 with 1 as the optimal value, it was still significantly higher than the other summaries. Despite the efforts to reduce hallucination through prompting and finetuning data, the LLM tends to produce more hallucinations than the other methods. This is at odds with other works researched that claim humans are more inclined to produce factual errors [68] [26]. On the other hand, this could be an artifact of the difficulties of detecting LLM-generated misinformation [87]. Considering hallucinations seem to occur quite infrequently for our model, users might be less prone and able to spot them. Such a scenario is definitely problematic for the end-product, and can not be tolerated. Xu et al. argue that although LLMs will never be free of hallucination, it is possible to make them learn some ground truth function, such as summarization [95]. Furthermore, they argue the key is to not regard LLMs as ground truth, but rather assistive tools for information retrieval. Therefore, output must be verified by human supervision. The verification can be alleviated with additional tools as SIG and Shapley values. In conclusion, hallucination is still problematic but can be reduced through further training and detected more easily with post-hoc methods.

Remarkably, the human summaries trends towards containing less false information than TextRank, which clearly can not be. Either, this is a statistical anomaly, or, there was a misunderstanding in what the question entails. However, considering the overall low results, and the expected result of the LLM making the most errors in the aspect, we believe the evaluators understood the question. The fact that at least for one instance, it was conceived that TextRank fabricated information displays the difficulty in evaluating such a question. Furthermore, it also conveys the extractive nature of TextRank was not entirely obvious to the evaluators.

5.6 Answer sample

As mentioned previously, the participants in the expert evaluation were volunteers through two steps. First, they had to sign up as interested and then actually complete the form. This creates two potential situations where selection bias can be introduced. Also, not all physicians at the hospital had the possibility to sign up in the first place. Out of those who did have the opportunity, we suspect the physicians who would benefit the most, the ones who spend the most time on administration, could not justify spending time on the activity when they have more pressing matters. Even though a randomized sample is preferred, we still think those samples would be the most helpful for the evaluation. On the other hand, just because they do spend such considerable time on administration, they could find it justified to spend 40 minutes on a task which could give greater benefits in the long run. However, we believe the low answer frequency demonstrates most could not find the time, despite their interest.

Comparing individual samples between evaluators, a clear discrepancy in scoring exists, especially in terms of fine-grain and scale usage. Notably, one evaluator scored the human summaries high in terms of overall, relevancy and missing information, but gave ones to both our model and TextRank for all instances. With the few number of evaluators in the study, it is difficult to determine whether such an opinion is more general in the surveyed population. As discussed, there is still a clear preference for the human summaries, but the other results did not deem our model and TextRank as completely useless, and we think there is a large difference between such results. We think the comparison between our model and TextRank also suffered, as their nuances were downplayed when a vastly better alternative existed.

6

Conclusion

Clinicians, and especially doctors, have some of the most challenging occupations in society, with dire consequences should they fail. We found that from an international perspective, Swedish doctors are more stressed and more likely to quit due in large part to their daily workload. While most of their work is allocated toward helping patients, as much as one-fifth may go to administration in the form of reading/writing discharge or admittance notes.

In this study, we explored the possibility of applying LLMs to generate summaries of Swedish electronic health records in an effort to reduce administration. We hypothesized it would be possible to achieve human-level performance on the task using an open-source model. We evaluated a number of models on the task and optimized the best candidate through techniques as prompt engineering, hyperparameter tuning and training.

Although the model did not manage to outperform the human, there are some key findings from the study. For the purpose of summarization, there is no need to employ the largest possible model, as smaller ones can be equally proficient at the task. Further, we find that a combination of high temperature and local typicality creates summaries which are better at mimicking human communication patterns. Finally, the feasibility of injecting trained LoRA weights on a different base model was observed, as it yielded the best results. This notion needs further study to find whether it is applicable in other scenarios as well.

Our model scarcely outperformed more classical approaches, but as it is possible to further adapt it to the downstream task, it is reasonable to assume the gap will only widen. On the contrary, LLMs pose a few risks that are not entirely redeemed by their advantages of automatic summarization, especially the black-box nature and tendency of hallucination. Hence, Swedish hospitals should tread carefully in the employment of LLM systems as substitutes for human information retrieval.

We hope that this thesis can contribute to furthering the discourse on using LLMs in a Swedish healthcare setting.

6.1 Applicability

If Sahlgrenska intends to implement our proposed system, there are still a number of challenges present. First and foremost, it has to be ensured sufficient hardware is

present in order to host the model with minimal downtime. Furthermore, it is not just our own model that needs hosting but also the model for the chat interface. As mentioned, Sahlgrenskas end goal is to use the summaries in RAG-fashion as input for the chat interface model. This model has not been decided yet, but assuming it is at least the same size, 3-4 of their 16 available GPUs would have to be preoccupied, limiting other research. This assumes only a single instance is hosted at the time. If the chatbot is to be of any help, it can not have latency issues tied to too many requests. Moreover, the initial effort to create all of the summaries would require an extensive amount of compute power and prioritization.

Alternatively, there is a possibility to use cloud computing. Towards the end of our project, there have been discussions that a legislative framework for allowing patient data to leave the premises could be in place within 2 years. Moreover, it would also allow for the usage of proprietary models which overall seem to perform better than open-source ones. For instance, we were not able to optimize our system to outperform GPT-4. After evaluating our model, and considering previous works within the field, we believe it to be a requirement in the pursuit of human-level performance.

Another consideration with using open-source models is their rate of improvement, as it can render the performance gains from our system obsolete. Our system will not become worse just because new models are released, but it does not seem reasonable to use a task-adapted model when a newer base model can simply outperform it. Meta released the third installment of the Llama models in late April 2024, with significant improvements over the previous installment. Considering they have been able to release three high-quality model installments since February 2023, all of them continually improving on the previous ones, it is not unreasonable to believe that a new model, by any developer, could be released within 6 months which would make a task-adapted Llama-3 obsolete as well. Obviously, this rate of improvement will not be feasible forever, but currently, it appears to be the case.

On the other hand, our project has given Sahlgrenska an optimization pipeline for the task where the model can simply be substituted. Yet, some steps in the pipeline would still require work when substituting a new model, e.g. prompt engineering, which is known to be brittle. The framework of DSPy was explored to this end, but we were not able to get any reasonable results [45]. Hence, the compute resources and maintenance required might still not be justified to change model and run the optimization pipeline continuously. However, as a pilot study for Sahlgrenska, it illustrates the resources required for an actual implementation, cloud-based or not. Whether our results are satisfactory or not, all the medical staff we have talked to have had a positive attitude towards the project. The physicians spoken to generally believe LLMs are a tool which would help them greatly due to the current administrative problems in Swedish healthcare. In fact, many have already experimented with the technology through ChatGPT.

That our study shows the difficulties for two people to construct such a system in four months, yet is still appreciated by the staff, shows more resources is needed for the task to render feasible results for Sahlgrenska. However, resources would be needed

not only to improve upon our findings but also for additional technical aspects, such as post hoc explanations. In this project, they have simply been brushed over as out of scope, but considering the EU AI act, they will be legal requirements.

The concerns which we have presented and the project itself only effects an implementation at Sahlgrenska, even though the issues of administration are present in all of Sweden. To address the problem nationally, a greater effort is required. First off, all units of care needs to be aligned on the current problem. As care units in Sweden consists of a mix between actors from the private and public sectors, interests can easily be disjoint. Currently, most Swedish hospitals work individually on implementing AI tools for care. This is justified as long as they are not trying to solve the same problem. However, this particular problem is known to exist across the country and is not limited to a specific region. Thus, to solve it for the entirety of Sweden, a greater initiative is required.

6.2 Limitations

The project was limited to focus on the creation of summaries and not their intended usage in the pipeline. Future work would have to include whether the summaries suffice as input for the chatbot. Even though they are not as of high quality as the human references, they could still be of sufficient quality.

The limitation of not being able to use patient data is justified, but still frustrating. It is completely reasonable patient documents are not allowed to be encoded into an LLM. However, we do believe a system using real data would outperform one based on synthetic, regardless of model or human-generated. It would give access to more data than what is reasonable to produce synthetically and they would also be samples from the actual distribution that is modeled. Still, the target summaries in their current form for the downstream task would not exist, but the time spent by physicians writing synthetic journals in this project could be directed to that instead. Alternatively, it could serve as a corpus for domain adaptation.

An additional vital consideration for further studies is to bridge the gap between the developers of the system and the physician end-users. This is always a general interest when creating any type of software, it should fulfill the requirements and expectations of the end-user. Nevertheless, in this case, the primary purpose of the system is to reduce time spent on administration. If the actual implementation can not achieve this, it does not matter how well the system performs. Moreover, this also applies to the development. The physicians whom have been part of the project have been of great help to us, yet it has not been trivial to align interests. Whereas we have been more focused on methodology and explorative research, the primary concern for them has indeed been results, which is reasonable. Our motivations have not always been justified to them, making them question intentions of our tasks. For the projects continuation, the gap needs to be bridged for better cooperation.

Bibliography

[1] C. Barkman and L. Aasa, *Onödig administration i sjukvården*, 2023. [Online]. Available: https://healthpolicy.se/rapport-onodig-administration/.

[2] McKinsey, *Tid till vård ger vård i tid*, Accessed: 2024-04-18, 2019. [Online]. Available: https://slf.se/nyheter/tid-till-vard-ger-vard-i-tid/.

[3] lagen. "Hälso- och sjukvårdslag (2017:30)." (Feb. 2017), [Online]. Available: https://lagen.nu/2017:30#K2P6S1 (visited on 05/06/2024).

[4] Myndigheten för vård och omsorgsanalys, *Vården ur primärvårdsläkarnas perspektiv - international health policy survey (ihp) 2022*, Accessed: 2024-04-17, 2022. [Online]. Available: https://www.vardanalys.se/rapporter/varden-ur-primarvardslakarnas-perspektiv-2/.

[5] G. Aronson and E. Bejerot, *Nödiga och oskäliga arbetsuppgifter bland läkare*, Accessed: 2024-04-21, 2012. [Online]. Available: https://lakartidningen.se/klinik-och-vetenskap-1/2012/11/onodiga-och-oskaliga-arbetsuppgifter-bland-lakare/.

[6] V. Götalandsregionen. "Uppdrag och vision." (Nov. 2022), [Online]. Available: https://www.sahlgrenska.se/om-sjukhuset/uppdrag-och-vision/ (visited on 04/24/2024).

[7] Göteborgsregionen. "Folkmängd i göteborgsregionen 2023." (Feb. 2024), [Online]. Available: https://goteborgsregionen.se/kunskapsbank/folkmangdigoteborgsregi
5.3d3d65dc17ee36e9de7ce73.html (visited on 04/24/2024).

[8] V. Götalandsregionen. "Kompetenscentrum ai datadrivet arbete i vården." (Feb. 2023), [Online]. Available: https://www.sahlgrenska.se/forskning-utbildning-innovation/samverkan/kompetenscentrum-ai/ (visited on 04/25/2024).

[9] OpenAI, *Gpt-4 technical report*, 2023. arXiv: 2303.08774 [cs.CL].

[10] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Systems with Applications*, vol. 165, p. 113 679, 2021, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113679. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420305030.

[11] R. Nallapati, F. Zhai, and B. Zhou, *Summarunner: A recurrent neural network based sequence model for extractive summarization of documents*, 2016. arXiv: 1611.04230 [cs.CL].

[12] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958. DOI: 10.1147/rd.22.0159.

[13] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, Jun. 1993, ISSN: 0891-2017.

[14] M. Banko, V. O. Mittal, and M. J. Witbrock, "Headline generation based on statistical translation," in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong: Association for Computational Linguistics, Oct. 2000, pp. 318–325. DOI: 10.3115/1075218.1075259. [Online]. Available: https://aclanthology.org/P00-1041.

[15] A. M. Rush, S. Chopra, and J. Weston, *A neural attention model for abstractive sentence summarization*, 2015. arXiv: 1509.00685 [cs.CL].

[16] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.

[17] M. Lewis, Y. Liu, N. Goyal, *et al.*, *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, 2019. arXiv: 1910.13461 [cs.CL].

[18] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, and M. Sun, *A unified model for extractive and abstractive summarization using inconsistency loss*, 2018. arXiv: 1805.06266 [cs.CL].

[19] X. Pu, M. Gao, and X. Wan, *Summarization is (almost) dead*, 2023. arXiv: 2309.09558 [cs.CL].

[20] H. Zhou, F. Liu, B. Gu, *et al.*, *A survey of large language models in medicine: Principles, applications, and challenges*, 2023. arXiv: 2311.05112 [cs.CL].

[21] R. Jain, A. Jangra, S. Saha, and A. Jatowt, *A survey on medical document summarization*, 2022. arXiv: 2212.01669 [cs.CL].

[22] S. Narayan, S. B. Cohen, and M. Lapata, *Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization*, 2018. arXiv: 1808.08745 [cs.CL].

[23] K. M. Hermann, T. Koiský, E. Grefenstette, *et al.*, *Teaching machines to read and comprehend*, 2015. arXiv: 1506.03340 [cs.CL].

[24] A. Fabbri, I. Li, T. She, S. Li, and D. Radev, "Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1074–1084. DOI: 10.18653/v1/P19-1102. [Online]. Available: https://aclanthology.org/P19-1102.

[25] K. R. McKeown, D. A. Jordan, and V. Hatzivassiloglou, "Generating patient-specific summaries of online literature," in *Papers from the 1998 AAAI Spring Symposium*, 1998.

[26] D. V. Veen, C. V. Uden, L. Blankemeier, *et al.*, *Clinical text summarization: Adapting large language models can outperform human experts*, 2023. arXiv: 2309.07430 [cs.CL].

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

[28] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL].

[29] O. Holmström, J. Kunz, and M. Kuhlmann, "Bridging the resource gap: Exploring the efficacy of English and multilingual LLMs for Swedish," in *Proceedings of the Second Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL-2023)*, N. Ilinykh, F. Morger, D. Dannélls, S. Dobnik, B. Megyesi, and J. Nivre, Eds., Tórshavn, the Faroe Islands: Association for Computational Linguistics, May 2023, pp. 92–110. [Online]. Available: https://aclanthology.org/2023.resourceful-1.13.

[30] T. Kew, F. Schottmann, and R. Sennrich, *Turning english-centric llms into polyglots: How much multilinguality is needed?* 2023. arXiv: 2312.12683 [cs.CL].

[31] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, P. Isabelle, E. Charniak, and D. Lin, Eds., Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. [Online]. Available: https://aclanthology.org/P02-1040.

[32] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013.

[33] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss, Eds., Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 65–72. [Online]. Available: https://aclanthology.org/W05-0909.

[34] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, *Bertscore: Evaluating text generation with bert*, 2020. arXiv: 1904.09675 [cs.CL].

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[36] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, *G-eval: Nlg evaluation using gpt-4 with better human alignment*, 2023. arXiv: 2303.16634 [cs.CL].

[37] S. Takeshita, S. P. Ponzetto, and K. Eckert, *Rouge-k: Do your summaries have keywords?* 2024. arXiv: 2403.05186 [cs.CL].

[38] N. Schluter, "The limits of automatic summarisation according to ROUGE," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, M. Lapata, P. Blunsom, and A. Koller, Eds., Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 41–45. [Online]. Available: https://aclanthology.org/E17-2007.

[39] C. Callison-Burch, M. Osborne, and P. Koehn, "Re-evaluating the role of Bleu in machine translation research," in *11th Conference of the European Chapter of the Association for Computational Linguistics*, D. McCarthy and S. Wint-

ner, Eds., Trento, Italy: Association for Computational Linguistics, Apr. 2006, pp. 249–256. [Online]. Available: https://aclanthology.org/E06-1032.

[40] J. Kasai, K. Sakaguchi, R. L. Bras, *et al.*, *Bidimensional leaderboards: Generate and evaluate language hand in hand*, 2022. arXiv: 2112.04139 [cs.CL].

[41] D. Muhlgay, O. Ram, I. Magar, *et al.*, *Generating benchmarks for factuality evaluation of language models*, 2024. arXiv: 2307.06908 [cs.CL].

[42] D. Huang, L. Cui, S. Yang, *et al.*, "What have we achieved on text summarization?" In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 446–469. DOI: 10.18653/v1/2020.emnlp-main.33. [Online]. Available: https://aclanthology.org/2020.emnlp-main.33.

[43] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, D. Lin and D. Wu, Eds., Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 404–411. [Online]. Available: https://aclanthology.org/W04-3252.

[44] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, *Large language models are zero-shot reasoners*, 2023. arXiv: 2205.11916 [cs.CL].

[45] O. Khattab, A. Singhvi, P. Maheshwari, *et al.*, *Dspy: Compiling declarative language model calls into self-improving pipelines*, 2023. arXiv: 2310.03714 [cs.CL].

[46] M. Caccia, L. Caccia, W. Fedus, H. Larochelle, J. Pineau, and L. Charlin, *Language gans falling short*, 2020. arXiv: 1811.02549 [cs.CL].

[47] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, *The curious case of neural text degeneration*, 2020. arXiv: 1904.09751 [cs.CL].

[48] C. Meister, T. Pimentel, G. Wiher, and R. Cotterell, *Locally typical sampling*, 2023. arXiv: 2202.00666 [cs.CL].

[49] J. Hewitt, C. D. Manning, and P. Liang, *Truncation sampling as language model desmoothing*, 2022. arXiv: 2210.15191 [cs.CL].

[50] A. Gupta, A. Shirgaonkar, A. de Luis Balaguer, *et al.*, *Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture*, 2024. arXiv: 2401.08406 [cs.CL].

[51] N. Houlsby, A. Giurgiu, S. Jastrzebski, *et al.*, *Parameter-efficient transfer learning for nlp*, 2019. arXiv: 1902.00751 [cs.LG].

[52] X. L. Li and P. Liang, *Prefix-tuning: Optimizing continuous prompts for generation*, 2021. arXiv: 2101.00190 [cs.CL].

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

[54] D. Kalajdzievski, *A rank stabilization scaling factor for fine-tuning with lora*, 2023. arXiv: 2312.03732 [cs.CL].

[55] N. Jain, P.-y. Chiang, Y. Wen, *et al.*, *Neftune: Noisy embeddings improve instruction finetuning*, 2023. arXiv: 2310.05914 [cs.CL].

[56] K. He, R. Mao, Q. Lin, *et al.*, *A survey of large language models for healthcare: From data, technology, and applications to accountability and ethics*, 2023. arXiv: 2310.05694 [cs.CL].

[57] J. Wei, X. Wang, D. Schuurmans, *et al.*, *Chain-of-thought prompting elicits reasoning in large language models*, 2023. arXiv: 2201.11903 [cs.CL].

[58] E. Commission, *Regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts*, Accessed: 2024-02-05, 2021. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206#document1.

[59] E. Parliament, "Artificial intelligence act: Deal on comprehensive rules for trustworthy ai," *Press Releases*, Dec. 2023. [Online]. Available: https://www.europarl.europa.eu/news/en/press-room/20231206IPR15699/artificial-intelligence-act-deal-on-comprehensive-rules-for-trustworthy-ai.

[60] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.," *Queue*, vol. 16, no. 3, pp. 31–57, Jun. 2018, ISSN: 1542-7730. DOI: 10.1145/3236386.3241340. [Online]. Available: https://doi.org/10.1145/3236386.3241340.

[61] J. Enguehard, *Sequential integrated gradients: A simple but effective method for explaining language models*, 2023. arXiv: 2305.15853 [cs.CL].

[62] H. Chen, I. C. Covert, S. M. Lundberg, and S.-I. Lee, *Algorithms to estimate shapley value feature attributions*, 2022. arXiv: 2207.07605 [cs.LG].

[63] A. Matthias, "The responsibility gap: Ascribing responsibility for the actions of learning automata," *Ethics and Information Technology*, vol. 6, no. 3, pp. 175–183, 2004. DOI: 10.1007/s10676-004-3422-1.

[64] T. Grote and P. Berens, "On the ethics of algorithmic decision-making in healthcare," *Journal of Medical Ethics*, vol. 46, no. 3, pp. 205–211, 2020, ISSN: 0306-6800. DOI: 10.1136/medethics-2019-105586. eprint: https://jme.bmj.com/content/46/3/205.full.pdf. [Online]. Available: https://jme.bmj.com/content/46/3/205.

[65] J. M. Durán and K. R. Jongsma, "Who is afraid of black box algorithms? on the epistemological and ethical basis of trust in medical ai," *Journal of Medical Ethics*, vol. 47, no. 5, pp. 329–335, 2021, ISSN: 0306-6800. DOI: 10.1136/medethics-2020-106820. eprint: https://jme.bmj.com/content/47/5/329.full.pdf. [Online]. Available: https://jme.bmj.com/content/47/5/329.

[66] E. COMMISSION, *Directive of the european parliament and of the council on adapting non-contractual civil liability rules to artificial intelligence (ai liability directive)*, Accessed: 2024-04-12, 2022. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52022PC0496.

[67] M. N. Duffourc and S. Gerke, "The proposed eu directives for ai liability leave worrying gaps likely to impact medical ai," *npj Digital Medicine*, vol. 6, no. 1, p. 77, Apr. 2023, ISSN: 2398-6352. DOI: 10.1038/s41746-023-00823-w. [Online]. Available: https://doi.org/10.1038/s41746-023-00823-w.

[68] L. Tang, Z. Sun, B. Idnay, *et al.*, *Evaluating large language models on medical evidence summarization*, 2023. [Online]. Available: https://doi.org/10.1038/s41746-023-00896-7.

[69] D. Cheng, S. Huang, and F. Wei, *Adapting large language models via reading comprehension*, 2024. arXiv: 2309.09530 [cs.CL].

[70] G. Adams, E. Alsentzer, M. Ketenci, J. Zucker, and N. Elhadad, "What's in a summary? laying the groundwork for advances in hospital-course summarization," *CoRR*, vol. abs/2105.00816, 2021. arXiv: 2105.00816. [Online]. Available: https://arxiv.org/abs/2105.00816.

[71] Z. Chen, A. H. Cano, A. Romanou, *et al.*, *Meditron-70b: Scaling medical pretraining for large language models*, 2023. arXiv: 2311.16079 [cs.CL].

[72] A. Ekgren, A. C. Gyllensten, F. Stollenwerk, *et al.*, *Gpt-sw3: An autoregressive language model for the nordic languages*, 2023. arXiv: 2305.12987 [cs.CL].

[73] A. Q. Jiang, A. Sablayrolles, A. Roux, *et al.*, *Mixtral of experts*, 2024. arXiv: 2401.04088 [cs.LG].

[74] L. Zheng, W.-L. Chiang, Y. Sheng, *et al.*, *Judging llm-as-a-judge with mt-bench and chatbot arena*, 2023. arXiv: 2306.05685 [cs.CL].

[75] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017, ISSN: 1091-6490. DOI: 10.1073/pnas.1611835114. [Online]. Available: http://dx.doi.org/10.1073/pnas.1611835114.

[76] T. Lesort, O. Ostapenko, D. Misra, *et al.*, *Challenging common assumptions about catastrophic forgetting*, 2023. arXiv: 2207.04543 [cs.LG].

[77] K. Gupta, B. Thérien, A. Ibrahim, *et al.*, *Continual pre-training of large language models: How to (re)warm your model?* 2023. arXiv: 2308.04014 [cs.CL].

[78] O. S. Initiative. "The open source definition." (Feb. 2024), [Online]. Available: https://opensource.org/osd (visited on 05/21/2024).

[79] D. Groeneveld, I. Beltagy, P. Walsh, *et al.*, *Olmo: Accelerating the science of language models*, 2024. arXiv: 2402.00838 [cs.CL].

[80] J. Kaplan, S. McCandlish, T. Henighan, *et al.*, *Scaling laws for neural language models*, 2020. arXiv: 2001.08361 [cs.LG].

[81] J. Hoffmann, S. Borgeaud, A. Mensch, *et al.*, *Training compute-optimal large language models*, 2022. arXiv: 2203.15556 [cs.CL].

[82] B. Sorscher, R. Geirhos, S. Shekhar, S. Ganguli, and A. S. Morcos, *Beyond neural scaling laws: Beating power law scaling via data pruning*, 2023. arXiv: 2206.14486 [cs.LG].

[83] AI@Meta, "Llama 3 model card," 2024. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[84] J. Öhman, S. Verlinden, A. Ekgren, *et al.*, *The nordic pile: A 1.2tb nordic dataset for language modeling*, 2023. arXiv: 2303.17183 [cs.CL].

[85] M. Malmsten, L. Börjeson, and C. Haffenden, *Playing with words at the national library of sweden – making a swedish bert*, 2020. arXiv: 2007.01658 [cs.CL].

[86] T. Zack, E. Lehman, M. Suzgun, *et al.*, "Assessing the potential of gpt-4 to perpetuate racial and gender biases in health care: A model evaluation study," *The Lancet Digital Health*, vol. 6, no. 1, e12–e22, 2024, ISSN: 2589-7500. DOI: https://doi.org/10.1016/S2589-7500(23)00225-X. [On-

line]. Available: https://www.sciencedirect.com/science/article/pii/S258975002300225X.

[87] C. Chen and K. Shu, *Can llm-generated misinformation be detected?* 2024. arXiv: 2309.13788 [cs.CL].

[88] I. Shumailov, Z. Shumaylov, Y. Zhao, Y. Gal, N. Papernot, and R. Anderson, *The curse of recursion: Training on generated data makes models forget*, 2024. arXiv: 2305.17493 [cs.LG].

[89] Y. Walter, "Artificial influencers and the dead internet theory," *AI & SOCIETY*, Feb. 2024, ISSN: 1435-5655. DOI: 10.1007/s00146-023-01857-0. [Online]. Available: https://doi.org/10.1007/s00146-023-01857-0.

[90] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, and A. Awadallah, *Orca: Progressive learning from complex explanation traces of gpt-4*, 2023. arXiv: 2306.02707 [cs.CL].

[91] H. Chen, X. Quan, H. Chen, M. Yan, and J. Zhang, *Knowledge distillation for closed-source language models*, 2024. arXiv: 2401.07013 [cs.CL].

[92] J. Wei, C. Yang, X. Song, *et al.*, *Long-form factuality in large language models*, 2024. arXiv: 2403.18802 [cs.CL].

[93] V. Fomenko, H. Yu, J. Lee, S. Hsieh, and W. Chen, *A note on lora*, 2024. arXiv: 2404.05086 [cs.LG].

[94] M. Aylett and A. Turk, "The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech," *Language and Speech*, vol. 47, no. 1, pp. 31–56, 2004, PMID: 15298329. DOI: 10.1177/00238309040470010201. eprint: https://doi.org/10.1177/00238309040470010201. [Online]. Available: https://doi.org/10.1177/00238309040470010201.

[95] Z. Xu, S. Jain, and M. Kankanhalli, *Hallucination is inevitable: An innate limitation of large language models*, 2024. arXiv: 2401.11817 [cs.CL].

A

Appendix 1

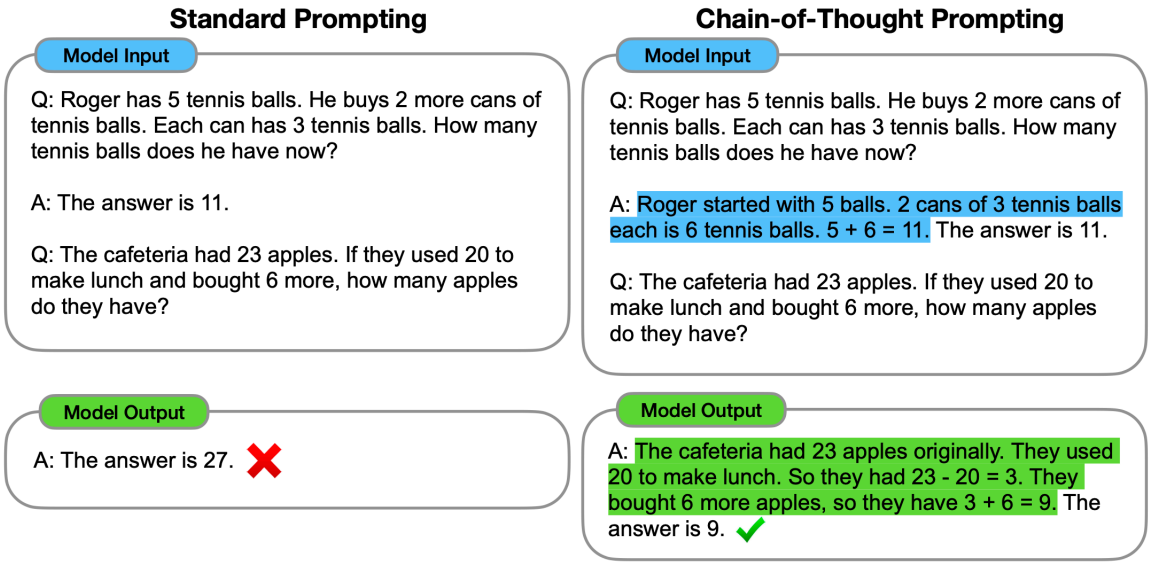

Figure A.1: An example of chain of thought prompting. The reasoning provided in the one-shot example makes the language model behave similarly, and improve their accuracy.
Source: [57]

```
[INST] <<SYS>>
You are a helpful medical assistant who helps medical profesionals by
    summarizing information about patients.
Answer with a text shorter than the original input.
Answer in Swedish.
Make sure the answer does not contain any introductory or pleasantry
    phrases.
<</SYS>>
Below are documents extracted from a patients medical journal.
<anamnesis>
*
</anamnesis>
Write a summary of the anamnesis in Swedish.
[/INST]
```

Listing A.1: The prompt used for recursive summarization

Notes	Operationsberättelse 24-02-03 kl 0900, Läk Dan Frölund: Operationskod: AAF05 VP shunt frontalt höger Strata 1,5 Operation: 5 Sahlgrenska Preoperativ bedömning: 79 årig man med konstaterad NPH, ca 1 års anamnes på tilltagande balansbesvär, bredspårig gång, urininkontinens, närminnespåverkan med MMSE 22 poäng. Bedömd på behandlingskonferens med indikation för shuntoperation. Anestesiform: Generell anestesi och lokalanestesi Operationsberättelse: Semicirkulär incision över Kochers punkt höger. Ett borrhål. Går till buken och tar upp ett växelsnitt ner till peritoneum. Kommer otvetydigt in i fri bukhåla. Tunnelerar därefter upp till ett hjälpsnitt bakom höger öra. Tunnelerar även från skalpsnittet till hjälpsnittet. En förinställd Strataventil (1,5) och distalkateter dras igenom i tvåsteg till buken. Kryssformad durotomi och kortikotomi. För ner en ventrikelkateter (längd [] cm). Klart utbyte. Kopplas på distala systemet. Verifierar distalt utbyte. Matar ner ca 40 cm kateter i fri bukhåla. Syr sedan igen skalpsnittet med inverterad enstaka Vicryl och fortlöpande Etilon. Enstaka etilon över hjälpsnittet. Vicryl i yttre muskelfascian och subkutant på buken, Etilon i huden på buken. Post-operativa ordinationer: Suturtagning 12 dagar post-op. CT kontroll 3 h post-op, remiss skrivs. Ingen mer antibiotika.
Summary	Sjukdomshistoria (Patientens diagnoser, sjukdomshistorik och riskfaktorer (t.ex. sjukdomar i familjen)): o NPH Sökorsaker (patientens symtom och/eller datum för ingrepp): o Balansbesvär o Bredspårig gång o Urininkontinens o Närminnespåverkan o MMSE 22 o Shuntoperation 24-02-03 Åtgärder (planerade undersökningar, behandlingar och åtgärder) o Post-op: o Suturtagning 12 dagar o CT kontroll: 3 h (remiss skriven) o Ingen antibiotika

Table A.1: Summary sample

Question	How does inhalation occur at rest?
Answer	Contraction of the diaphragm and external intercostal muscles increases the volume of the lungs - the increase in lung volume leads to a decrease in pressure, allowing air to enter the lungs - the lungs are elastic (expanded).

Table A.2: Question and Answer sample translated to English.

Model	ROUGE-L	BLEU	METEOR
Open Source			
`meta-llama/Llama-2-13b-chat-hf`	**0.351**	**0.163**	**0.383**
`meta-llama/Llama-2-70b-chat-hf`	0.343	0.078	0.317
`malhajar/meditron-70b-chat`	0.138	0.032	0.122
`AI-Sweden-Models/gpt-sw3-20b-instruct`	0.301	0.102	0.291
`Falconsai/medical_summarization`	0.104	0.104	0.088
`mistralai/Mixtral-8x7B-Instruct-v0.1`	0.290	0.051	0.230
Rule Based			
`summa-textrank`	**0.171**	**0.058**	**0.130**
Proprietary			
`openai/gpt-4-0125-preview`	**0.515**	**0.266**	**0.488**
`openai/gpt-35-turbo-16k`	0.366	0.131	0.310
Model Tailoring			
`Llama-2-13b-chat-hf` + Best Gen-Config.	0.392	0.195	0.414
DA	0.03	0.01	0.05
DA + QA	0.10	0.03	0.07
QA	0.328	0.085	0.319
Model 1	0.252	0.066	0.346
Model 2	0.347	0.112	0.349
Model 3	0.388	0.145	0.375
Model 2 LoRA loaded on base model	0.398	0.206	0.431
Model 2 LoRA loaded on base model + Best config	**0.435**	**0.225**	**0.466**

Table A.3: Comparison of all models and variants. Higher is better across all metrics, best score is bold, $n = 24$.

Config	ROUGE-L	BLEU	METEOR	BERTScore
Best	0.435	0.225	0.466	0.770
Worst	0.341	0.160	0.379	0.739

Table A.4: Table showing the difference in performance between the best and the worst hyperparameter configurations, and that they correlate with BERTScore.