BHT Berliner Hochschule für Technik

De-identification of clinical report data through natural language processing with large language models

vorgelegt von

Daniel Schruhl

Matrikel Nr.: 910754

dem Fachbereich Fernstudieninstitut der BHT
der Berliner Hochschule für Technik Berlin
vorgelegte Masterarbeit
zur Erlangung des akademischen Grades

Master of Science (M.Sc.)

im Studiengang

Medizinische Informatik

Tag der Abgabe February 9, 2024

Studiere Zukunft

Gutachter

Dr. med. Markus Bockhacker — Helios Kliniken GmbH
Prof. Dr.-Ing. habil. Alexander Löser — Berliner Hochschule für Technik

Kurzfassung

Diese Masterarbeit vergleicht Llama-2, Falcon und Mistral im Hinblick auf die Pseudonymisierung von medizinischen Laboraufzeichnungen. Es wurden verschiedene Prompts und LLM Konfigurationen evaluiert, um ihren Einfluss auf die Pseudonymisierung zu messen. Mistral machte bei dieser spezifischen Aufgabe weniger Fehler als die anderen LLMs, jedoch wurden immer noch Fehler bei der Identifizierung von Personen und deren Namen gemacht. Diese Masterarbeit schlägt außerdem Prompts für eine solche Aufgabe vor und zeigt die Grenzen von LLMs auf, indem sie ihr Verständnis von Konzepten wie „Personen“ verdeutlicht. Es wird unter anderem die Bedeutung der Diversität der Datensätze diskutiert, die zum Training von LLMs verwendet werden, sowohl im Bezug auf die Auswahl der Datensätze als auch auf den Kontext innerhalb der Datensätze.

Abstract

This thesis compares Llama-2, Falcon, and Mistral in de-identifying medical laboratory notes. Various prompt formats and LLM configurations were evaluated to determine their impact on the de-identification process. Mistral made fewer errors for this specific task than the other LLMs, but still made mistakes in identifying people and their names. This thesis provides guidance for creating prompts for such a task and highlights the limitations of LLMs, clarifying their understanding of concepts such as "people". The thesis also emphasizes on the importance of diverse datasets used to train LLMs, both in terms of dataset selection and context within the datasets.

Erklärung

Ich versichere, dass ich diese Abschlussarbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Datum Unterschrift

Contents

List of Figures

List of Tables

Chapter 1

Motivation

1.1 Introduction into the problem space

Helios IT Service GmbH operates the IT infrastructure of 87 hospitals that serve 5.5 million patients yearly [hel 2023]. These hospitals collect patient-related information such as procedures received, movements between departments and rooms, and other relevant data. This is done for quality assurance and accounting purposes. The data a patient produces in the hospital and its IT systems are personal data as defined by Art. 9 GDPR[1]. Processing it comes with strict requirements.

This data has tremendous potential to improve patient care and the overall hospital services. It could be used to optimize confidence of lab results or clinical tests, further strengthening the decision process of which lab tests to perform and when, and reducing the time lost to unnecessary testing. It could also be used to gain insights into which treatments can lead to higher chances of success, optimizing for fewer misdiagnoses, reducing errors in patient care or machine learning-based decision making in time-sensitive environments (e.g. emergency room scenarios) that ultimately impact the chance of survival.
While the benefits of using this data to improve patient care and treatment in the hospital are clear, making it usable is very challenging. Mainly because of the personal nature of such data. For example, during patient care, hospital staff add notes to each procedure performed. These notes contain a variety of personal or personally identifiable information, such as names, locations, and more.

Identifying named entities (named entity recognition) has been researched since the 1990s according to [Liu et al. 2022] and has been continuously researched since then [Wang et al. 2018] [Tseo et al. 2020]. However, doing so in clinical contexts poses different challenges. Clinical texts contain uncommon words compared to other texts (e.g., Latin names of diseases, names of drugs), abbreviations, codes (e.g., ICD-10 to encode diseases or treatments), or medication information (e.g., units, numbers). In this context, we are looking at notes that are not necessarily sentences. They may also contain names based on citations of medical literature or procedures. All this leads to further ambiguity in the texts (see Figure 1.1).

Building a rule-based system to identify named entities in these texts requires a lot of expert knowledge (technical and linguistic). It also requires the development of a complex system that covers many edge cases in order to work correctly on the entirety of the given data. On first glance, a probabilistic approach with generalization seems to be suitable, because less expert domain knowledge is required and the development is less complex in terms of encoding rules

[1] https://web.archive.org/web/20240113211916/https://gdpr-info.eu/art-9-gdpr/

correctly and in their entirety. However, a probabilistic model carries risks, especially with regard to false negatives. One false negative would be enough to violate data privacy and safety rules and make the use of such a system impossible. On the other hand, training a neural network from scratch for this specific task is time-consuming. Using a pre-trained large language model (LLM) can speed up this process significantly.
This thesis investigates the conditions under which circumstances an LLM can perform named entity recognition for names of individuals in clinical notes in a German corpus, in order to make the notes accessible for further use cases.

> "Laut aktueller Literatur liegt bei einem Wert von >20 µg/l ein ausreichender Vitamin D-Spiegel für eine normale Knochenmineralisation vor, Werte zwischen 12 µg/l und 20 µg/l sprechen für eine Insuffizienz, Werte <12 µg/l für einen Vitamin D-Mangel. Toxische Spiegel werden ab Werten >100 µg/l erreicht. Hinweis zur Umrechnung: 1 µg/l = 2,5 nmol/l. [Munns CF et al: Global Consensus Recommendations on Prevention and Management of Nutrinional Rickets. J Clin Endocrinol Metab. 2016;101:394-415])"

Figure 1.1: One example taken from the existing data set. It contains numbers and units, a source for reference and medical vocabulary. The sentence shows how different the underlying data is to commonly found text corpus that do not belong to the medical domain. This example contains no name of a person that needs to be retracted (the name in the source is to be ignored).

1.2 Related work

As previously mentioned, named entity recognition is a well-established research field in natural language processing (NLP). State-of-the-art (SOTA) models, when combined with rule based systems, can accurately identify named entities [Abu-El-Rub et al. 2022]. However, it is important to note that these models are typically trained and evaluated using English datasets [Lample et al. 2016]. Additional research indicates that the performance of SOTA models varies significantly depending on the context of the datasets used, such as news, telephone conversations, and magazines [Vajjala und Balasubramaniam 2022]. The datasets SOTA models are trained and evaluated on, often do not include medical texts [Yamada et al. 2020]. Using SOTA models for a German medical dataset may not yield the same level of performance as suggested by the model's evaluation.

Custom-built and trained neural networks have been used for de-identification tasks on medical notes, but transformer architectures have not been successful in this area[Ahmed et al. 2021]. Research has primarily focused on training conditional random fields, which lead to promising results, even for German medical notes [Seuss et al. 2017]. Similarly, bidirectional long short-term memory (LSTM) networks have achieved positive outcomes for German medical notes [Richter-Pechanski et al. 2019]. However, all of these approaches require custom neural networks to be engineered and trained, which is a time-consuming and laborious process.

Thus far, research in the medical context has focused on either using pre-trained neural networks in combination with another system or training a custom model from scratch. Notably, none of these approaches have utilized only a pre-trained model for the purpose of de-identification in medical texts. This is likely due to the fact that most SOTA pre-trained models are trained using non-medical text data. Utilizing pre-trained models can significantly speed up the overall

development process of delivering such a feature in real-life scenarios, as it eliminates the engineering and evaluation effort required for a custom-built neural network. It also eliminates the need for a large amount of training data.

LLMs can be used for a variety of NLP tasks with vastly different contexts [Touvron et al. 2023a]. They can even be used for medical texts [Han et al. 2023]. Modifying the prompt of an LLM can determine the task for which it is used, e.g. named entity recognition [Ashok und Lipton 2023]. In addition, the prompt can be used to do one-shot, few-shot or even zero-shot learning [Brown et al. 2020] [Sanh et al. 2021]. LLMs can speed up the development process, but engineering the right prompt significantly affects the overall performance and task the LLM performs [Lester et al. 2021]. Unfortunately LLMs require a large amount of GPU resources to run inferences. Fortunately, methods such as Low-Rank Adaptation (LoRA) and quantization have been devised to reduce the performance restrictions of LLMs, making them much more efficient to run on CPUs [Hu et al. 2021] [Shen et al. 2023].

Pre-trained LLMs can function as multilingual models, but their performance on languages other than English is significantly lower, as noted by [Zhang et al. 2023]. This, combined with the vast differences in vocabulary found in medical domain texts, highlights the difficulty of the task at hand. While LLMs that have been trained on medical domain datasets such as MedAlpaca do exist, it is important to note that they have been fine-tuned on datasets that are predominantly in English [Han et al. 2023]. Moreover, English medical vocabulary differs from German medical vocabulary.

Other research utilized publicly available medical domain texts. These datasets can result in models that do not perform as effectively in real-life scenarios. This is due to the fact that the used datasets do not show similar distributions of tokens in a vocabulary, that occur in environments outside of research. For instance, there is research that leverages the MIMIC-III dataset [Grundmann et al. 2021] [van Aken et al. 2022]. This dataset contains texts with ICD-9 classifications. German hospitals currently use ICD-10 (with modifications) [bfa 2023]. Other studies have demonstrated that publicly available datasets for medical purposes can yield models unsuitable for clinical use [Roberts et al. 2021]. In many instances, this is attributed to poor quality of data, as these datasets differ in their distributions to actual data in clinical contexts. In some cases even leading to biases. On top of that, the datasets are often also mislabeled [Wynants et al. 2020].

1.3 Proposed solution

This thesis is using LLMs to leverage the advantage of not having to train a custom neural network or coming up with an expert system to identify names of people in medical texts. Several LLMs are available for this purpose. In particular, Falcon [Almazrouei et al. 2023], Llama-2 [Touvron et al. 2023b] and Mistral [Jiang et al. 2023] are going to be examined. As mentioned above, prompt engineering is critical for the output of the LLM. For this purpose, multiple prompts are compared to determine the best one for the task of identifying names of individuals. These include, for example, zero-shot, one-shot and few-shot prompts with different levels of information and structures. Additionally, the LLMs have to run on a CPU. This is done using quantization

The proposed solution consists of a de-identification API that exposes the LLM through a REST API (using Flask[2]) and a de-identification library that provides a PySpark UDF[3] that can call the de-identification API (see Figure 1.2). The UDF from the de-identification library can then be called from within PySpark code inside data pipelines to transform DataFrames within the Spark cluster.

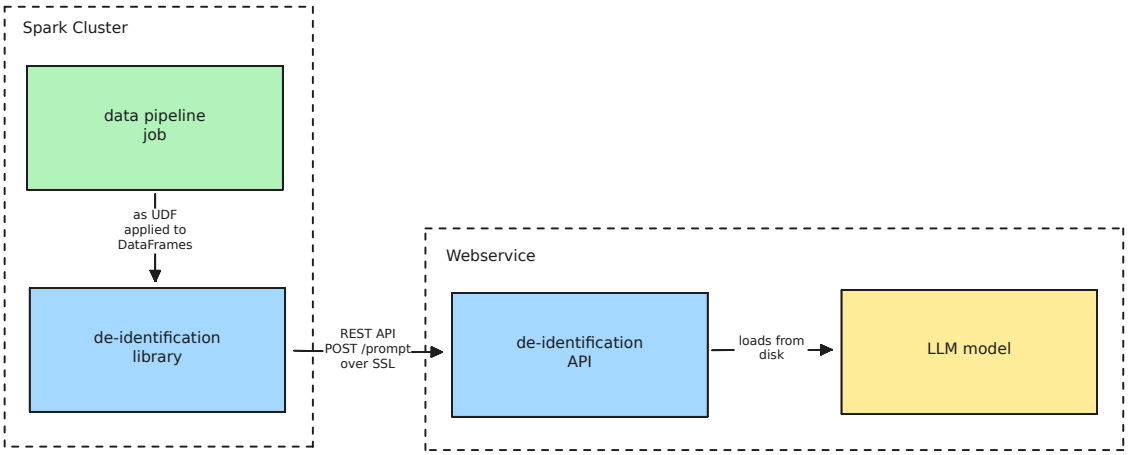

Figure 1.2: Overview of components of proposed solution. Blue boxes are components this thesis created. The LLM model will be exposed through a REST API. A library will access the deployed REST API through a PySpark UDF that can be applied on DataFrames within data pipeline jobs that run in a Spark cluster.

The de-identification API is served through a gunicorn HTTP server[4]. This way it can handle production work loads. It is exposed through an encrypted connection (TLS) and deployed within the internal network the Spark cluster can access. It is deployed once in the environment so all consequent Spark worker nodes can call the just once initialized LLM through the API instead of initializing it per UDF call (see Figure 1.3). The de-identification API abstracts wrapping incoming requests of clinical lab notes inside a prompt that can extract the names of people from within that note. The REST API then returns the names of the individuals. It uses the pre-configured LLM that is locally available in the de-identification API and loaded into memory at runtime.

The de-identification library provides a UDF that is calling the de-identification API. It uses production-ready techniques like timeouts and retries with delays for the calls to the API to accommodate for the fallacies of distributed computing[5]. The library replaces the names returned by the API with redaction characters (xxx) for all text within a column in a DataFrame. The UDF makes sure the transformations are run on the worker nodes (see Figure 1.3) in the Spark cluster rather than the driver nodes (for simplification reasons called "driver nodes", in reality a combination of driver program and cluster manager). The library is provided as a wheel inside the Spark cluster. This allows Spark jobs running inside a data pipeline to leverage this functionality.

[2] https://web.archive.org/web/20240119025911/https://flask.palletsprojects.com/en/3.0.x/

[3] https://web.archive.org/web/20220819103426/https://spark.apache.org/docs/3.1.1/sql-ref-functions-udf-scalar.html

[4] https://web.archive.org/web/20240119054207/https://gunicorn.org/

[5] https://web.archive.org/web/20171107014323/http://blog.fogcreek.com/eight-fallacies-of-distributed-computing-tech-talk/

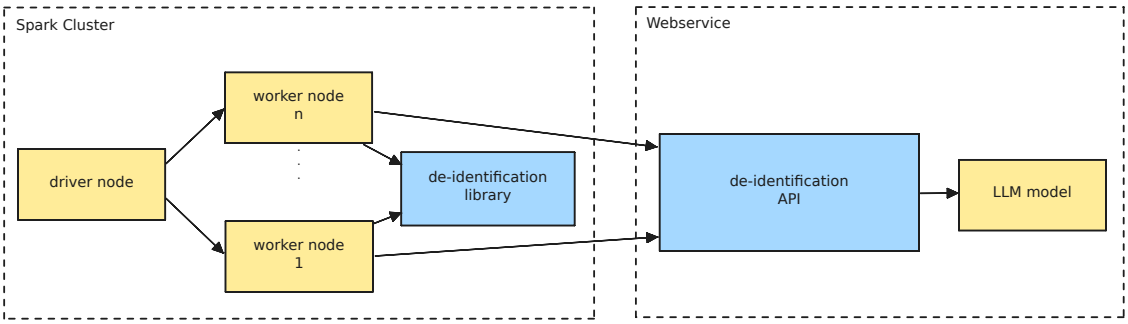

Figure 1.3: Simplified representation of the distribution of the components over infrastructure components inside the Spark cluster. The de-identification API is loaded and deployed once, giving the Spark workers the chance to call it in parallel.

The proposed solution needs to run on-premise in a compute cluster for data privacy and security purposes. The cluster itself does not have internet access. Therefore the LLM needs to be shipped together with the de-identification API through a docker container. This is done during the docker image build process. The compute cluster is CPU-based, which requires the LLM to do inference on a CPU. This is accomplished through the use of llama-cpp-python[6] and LangChain[7].

All components were developed in Python while applying TDD[8] and adhering to the test pyramid[9]. The code is available as open source at `https://github.com/meandor/huecaya` under the MIT license.

[6] `https://web.archive.org/web/20240106153614/https://github.com/abetlen/llama-cpp-python`

[7] `https://web.archive.org/web/20240106120236/https://python.langchain.com/docs/integrations/llms/llamacpp`

[8] `https://web.archive.org/web/20240104021218/https://martinfowler.com/bliki/TestDrivenDevelopment.html`

[9] `https://web.archive.org/web/20240115163428/https://martinfowler.com/bliki/TestPyramid.html`

Chapter 2

Existing Data

2.1 Exploratory data analysis

The dataset provided by Helios contains 852,514 rows of German hospital lab notes. The lab primarily conducted microbiology and clinical chemistry tests. These notes may contain critical information that can aid in the interpretation of lab results, the process leading to the results, or additional patient care information. This may constitute critical information that could facilitate the overall recovery process for patients. The notes not only contain names of people but also locations, times, and internal identification numbers. All of these can be used with different efforts to identify a person (especially in combination), making them personally identifiable information (PII). To demonstrate the feasibility of using LLMs to de-identify these notes, the scope of this thesis was limited to only names.

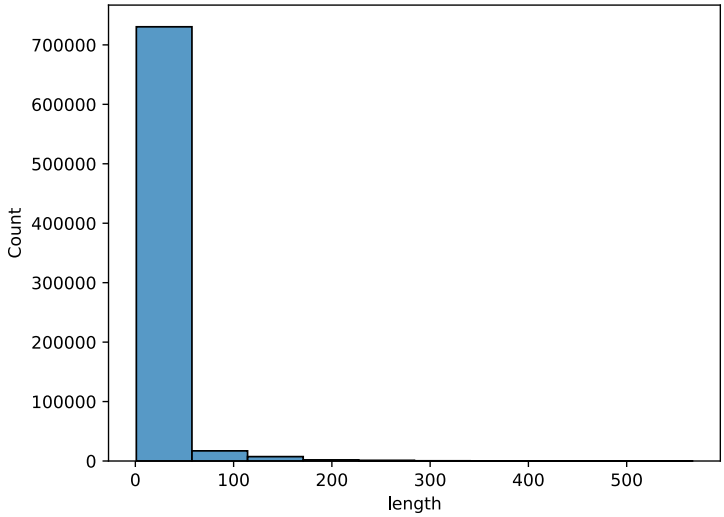

Figure 2.1: Histogram over the cleaned dataset using 10 bins. The first bin is the biggest by far, indicating that there might be outliers or a bias.

The data is comprised of strings that do not necessarily need to have content or form complete sentences. The dataset was cleaned by removing empty rows, which, in this case, refers to rows that are truly empty, empty strings, or are composed solely of whitespaces. In total, 93,686 rows were removed, representing approximately 11% of the raw data. The dataset contained a substantial number of duplicates due to the use of multiple notes conveying the same meaning. For instance, there were numerous rows consisting solely of `"----------"`, indicating an absence of content for that note.

In the further analysis of the cleaned dataset, the word count for each text was calculated. The counting process was simplified by splitting the notes on each whitespace, with each separated element being considered a word. The word count within the notes ranged from 1 to 567 words. The mean of the number of words within the notes is 14.47 and 75% of the data has the note length of 13 words or less. This relatively low number suggests that the majority of notes are shorter rather than longer, and many notes may lack useful information. This can be confirmed by examining the distribution of the note lengths in Figure 2.1. Further analysis into the notes containing 60 words or less (first bin) shows that there are two big outliers at notes with 5 and 7 words (see Figure 2.2).

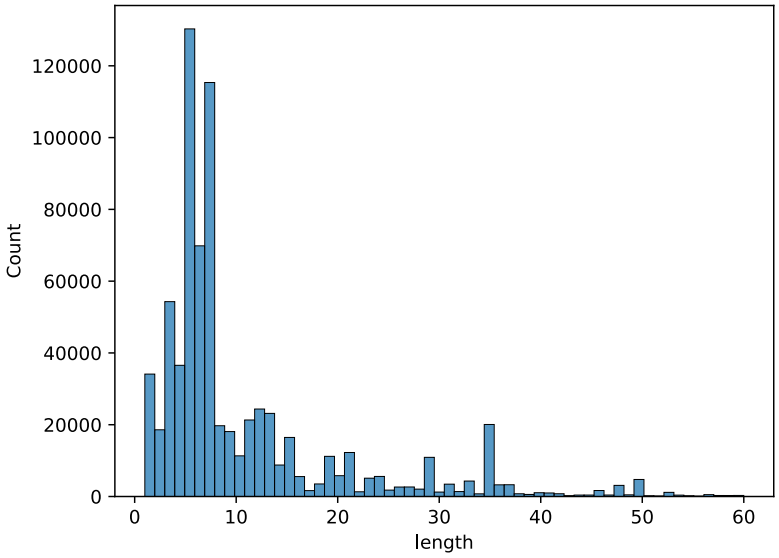

Figure 2.2: Histogram of the notes that have 60 or less words. Each length is its own bin (60 bins). There are two outliers visible.

Random samples of notes with fewer than 8 words were specifically reviewed. The samples were checked for the string "Dr.", indicating the presence of medical personnel in the note. The samples mostly consisted of sentences of the sort or similar to:

> "von Dr. Jane Doe an Station"

These are notes that hint at who initiated and who performed the lab testing. The informational use of these notes for improving patient care and recovery is very low. In comparison, notes of the size of 8 words or more were analyzed in a similar manner and contained sentences like (manually obfuscated for anonymization):

> "Hinweis: reduziert 13/37 Grund: Rheuma (bei Dr. John Doe)"

> "Hinweis: umtägig im Wechsel 42 mg und 4,2 mg, Fr. Dr. Jane Doe, Rheumatologie"

> "siehe Text Siehe Text: Material : Nasen-Rachenabstrich (Screening) Abnahme : 04.02.2023 13:37 interne Nummer: 133742 Mikrobiologischer Endbefund MRSA (kulturell): nicht nachgewiesen Befundverantwortlich Dr. Doe"

These notes include medication information, lab results and medical diagnoses. The information they contain is much higher and poses therefore more usefulness for improving patient care and recovery. Because of this, the dataset was further filtered for notes that contain 8 words or more.

Figure 2.3 shows a much better distribution over the data in terms of length of the notes after the filtering was done. In total, the dataset was reduced to 299,795 notes, which represents 35% of the raw dataset. The mean is now at 29.1 and 75% of the data has the length of 33 words or less. This overall distribution shows less bias towards a specific note structure which helps in determining the generalization performance of the LLMs. The distribution still shows a majority at notes with less than 22 words but this seems to be more of the actual nature of the notes that people write in this context and that are useful.

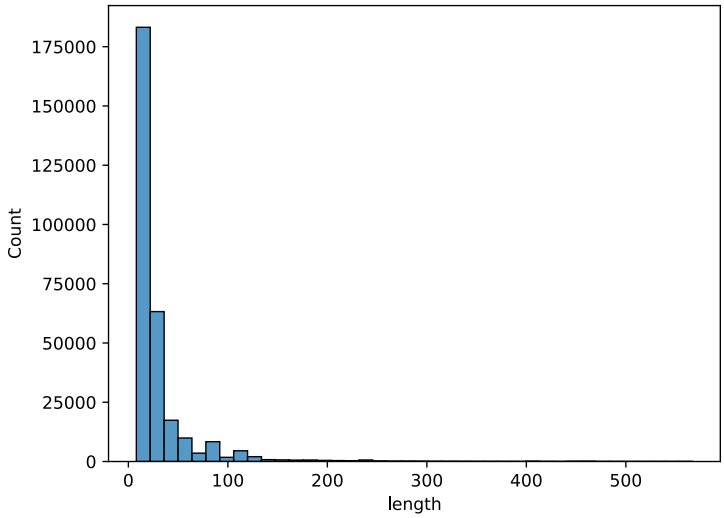

Figure 2.3: Histogram of the notes that have 8 or more words. 40 bins were used.

2.2 Selecting a dataset for further use

The previously cleansed dataset was randomly shuffled. The first 700 rows were manually labeled and selected for further use. The underlying distribution of the cleansed dataset can be found in the test dataset (see Figure 2.4). Because of the random shuffling, there are some small differences visible in the histogram but the overall representation of the original distribution is visible in the test dataset.

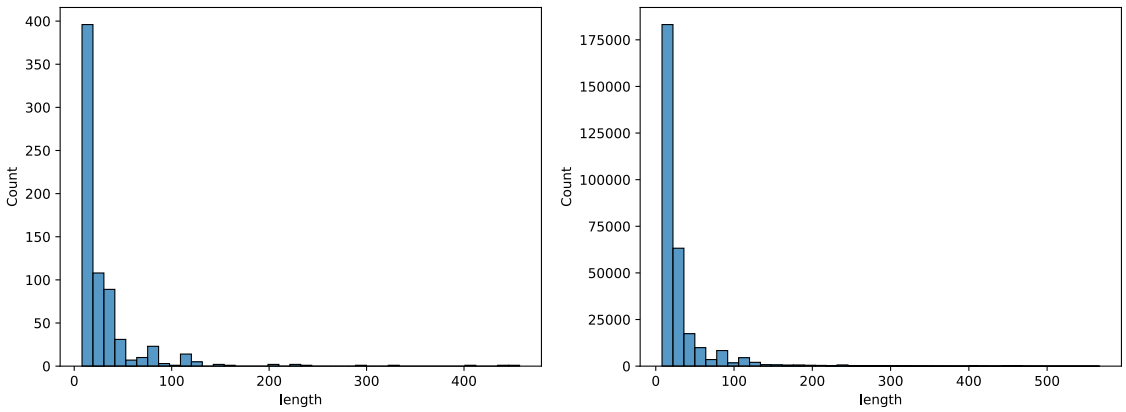

Figure 2.4: Histogram of the randomly selected test data (left) in comparison to the completely cleaned dataset (right). 40 bins were used for both.

44 names are inside the 700 rows. Some rows only had one name in them, some had multiple. There was no pattern in their distribution visible. The data was labeled with either an empty string for no name being present or the names that were present separated by a comma and a whitespace. For example: "Prof. Dr. med. Jane Doe, Dr. med. John Doe". The names of authors from citations were skipped, as they are not considered PII data. One label occurred twice, while the others were unique.

Chapter 3

Method

3.1 Used models and their parameters

At the time this work was conducted, only a handful of open source models existed. Among them, Llama-2 stood out as one of the most widely adopted and integrated models within the ecosystem. Apart from that, Falcon emerged as its closest competitor in terms of performance, according to the Hugging Face open LLM leaderboard[1]. Mistral was just released and showed promising results. For the purpose of this thesis, only the 7B parameters version of all the models were used, as these were running with an acceptable inference time on the given hardware.

These models come with their pre-trained parameters (weights and biases) and activations (intermediate values being generated during the forward pass of the neural network). They can be quite large (around 30gb for Llama-2), which may pose challenges loading them into memory (RAM or GPU memory) due to limited available space and computational resources. Quantization is one way to overcome this and make the models usable in CPU and memory restrained contexts. When the models are trained, their parameters and activations are stored as 32bit floating-point numbers. Quantization describes the process of reducing these numbers from 32bit floating-point numbers to smaller bit integers. As the numbers go down from 32bit floating-point to, for example, 8bit integers, they loose precision due to the limited memory available to store them. This leads to quality losses in the overall performance of the neural network. However, there is a significant reduction in the required computational and memory resources.

All models were used in a quantized format, in order to enable efficient inference on the given CPU. Q5_k, Q4_K and Q5_1 were used for quantization. Only Falcon was used with a Q5_1 quantization type, as no existing pre-trained and quantized model of Falcon existed with Q5_k quantization at that point in time. Q4_K was used to get to results faster, to test the overall feasibility. The Q5_k quantization type was used because it balances the tradeoff between speed and performance. It is not too large, so it will perform inferences in an acceptable time, but at the same time it is not too compressed, so it still yields an acceptable performance.

Q5_1 and Q5_k quantization both use a similar approach to quantization. They both convert a 32bit floating-point number into a 5bit signed integer. They do so by leveraging following equation:

$$q = \lfloor \frac{w - m}{d} \rceil, \text{where} \tag{3.1}$$

[1] `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, visited 2023-10-12

q is the quant
w is the original weight
m is the block minimum
d is the block scale

The block scale d is calculated with:

$$d = \frac{w_{max} - w_{min}}{2^b}, \text{ where} \tag{3.2}$$

w_{max} is the max weight inside the block
w_{min} is the min weight inside the block
b is the bits that are used for quantization

What is then stored in place of the floating point numbers of the weights are the quants q as 5bit signed integer values. This reduces the memory footprint significantly, which decreases the performance needed and enables inference on a CPU. At inference time the quants are used to calculate the weights w' with:

$$w' = d \cdot q + m, \text{ where} \tag{3.3}$$

w' is the weight
d is the block scale
q is the quant
m is the block minimum

Since the quants are based on integer values and calculated using rounding (see Equation 3.1), reconstructing the weight w' leads to a precision loss of the original weight w.
Q5_1 quantization goes over the entirety of all weights. Let's assume Llama-2 has weights $w \in [-2.0, 2.0]$, it would mean: $m = -2.0$, $d = \frac{2.0+2.0}{2^5} = 0.125$. These values can be used to calculate the quant from each weight and the reconstructed weight from each calculated quant.

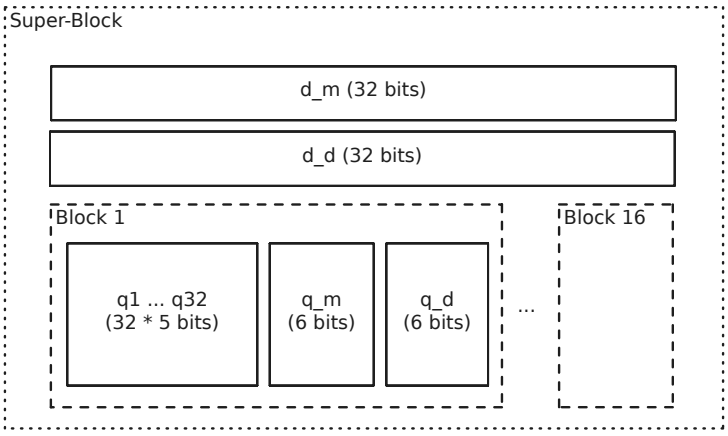

Figure 3.1: Q5_k structure of one super-block containing 1 scale for quantized scales (d_d), 1 scale for quantized minimums (d_m) and 16 blocks. Each block containing 32 quantized weights ($[q_1, ..., q_{32}]$) and 1 quantized minimum (q_m) and 1 quantized scale (q_d). To calculate the weight w'_1 from block 1: $w'_1 = q_1 \cdot q_d \cdot d_d + q_m \cdot d_m$.

Q5_k quantization goes further than Q5_1 quantization by creating groups of weights as so called "blocks". One block holds 32 weights. 16 of these blocks are grouped together as "super-blocks" (see Figure 3.1). The above equations hold true for each block. Each block has one block minimum (m) and one block scale (d), leading to 16 block minimums and 16 block

scales for each super-block. These values are quantized as 6bit integers for each super-block. Therefore, each super-block holds an additional super-block scale for the quantized block scales and a super-block scale for the quantized block minimums (both 32-bit floating point - full precision). The quantization for the block minimum and the block scale is done without a minimum m.

By doing so, a weight effectively utilizes 5.5 bits[2]. The block wise quantization aids in maintaining a higher degree of precision because the scale potentially holds fewer values and can be more fine granular. The grouping into super-blocks helps in the compression effort [3].

Both Mistral and Falcon are also available with fine-tuned models based on an instruction specialized dataset. These models were also used to further determine their capabilities (see Table 3.1).

Name	Parameter size	Quantization
Llama-2	7B	Q5_k
Falcon	7B	Q5_1
Falcon (instruct)	7B	Q5_k
Mistral	7B	Q4_k
Mistral (instruct)	7B	Q5_k

Table 3.1: The utilized models with their parameter size and used quantization.

The models can be further adjusted with additional parameters. Various values for context length, maximum answer length (max. length) and the temperature have been used (see Table 3.2).

The context length refers to the number of tokens the LLM considers as input when generating predictions for the next tokens, while the max. length describes the maximum length of the tokens the LLM generates as output.

The temperature describes the randomness of the LLM's output. A temperature closer to 0 will produce a more deterministic behaving output, whereas a temperature closer to 1 can create outputs that differ for the same input (often referred to as exploitation vs. exploration in reinforcement learning settings) [He et al. 2018].

Context length	Max. length	Temperature
$[1024, 2048]$	$[50, 100, 250]$	$[0.0, 0.5, 0.9]$

Table 3.2: Initial parameters used for each model in a grid search to determine the best model and parameters. Creating a grid out of these parameters will create 18 combinations ($2 \cdot 3 \cdot 3$).

The specific values in Table 3.2 were chosen for covering the available ranges for each parameter. The temperature has a range of $[0.0, 1.0]$. The chosen values represent two extremes (0.0 and 0.9) and one value in the middle (0.5). The same strategy was employed for the other parameters. The context was chosen to be $4096/2$ at its maximum value, to examine the impact of a restricted context (the lowest available context in the models is 4096). Further specific parameter and prompt combinations were examined in greater detail based on the results of these parameters.

[2] $(16 \cdot 32 \cdot 5 + 2 \cdot 32 + 2 \cdot 16 \cdot 6)/16 \cdot 32 = 5.5$

[3] https://web.archive.org/web/20240106060008/https://github.com/ggerganov/llama.cpp/pull/1684

3.2 Prompts

As mentioned earlier, different prompts were used. Completely German prompts were used and prompts with English instructions and German notes. These prompts employed zero-shot, one-shot and few-shot learning techniques, as well as a structure known as PromptNER [Ashok und Lipton 2023]. The prompts were tested out both with and without context paragraphs. All prompts were anonymized and contain fictitious names and numbers. To separate examples in the prompt, the string "###" was used. The LLM picked this up and used it in its output to distinguish between multiple answers. The prompts were managed through a prompt template that included the prompt and a placeholder string "{sentence}" to be replaced with the actual note from the test dataset (see Figure 3.2).

> You are an expert in linguistics and extract people with names from texts. A name can consist of a firstname and a lastname or of multiple of these. Names can also consist of titles. Extract from the following sentence the people.
> ###
> Sentence: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
> People: Paola Mertens, Peter Worms
> ###
> Sentence: '{sentence}'
> People:

Figure 3.2: English one-shot prompt with context and instruction. It has one example.

The exact prompts can be found in the Appendix A and B. For context, the prompts used in the study can be found in Table 3.3. Each prompt has a new name by which they will be further referenced with. Some prompts were used for fine tuning with specific parameters and only for certain models. Those prompts are highlighted in Table 3.3 (indicated with $*$).

Prompt name	Reference
zero-shot-german-with-context	figure A.1
3-few-shot-german	figure A.2
3-few-shot-german-with-context	figure A.3
prompt-ner-german	figure A.4
zero-shot-english	figure B.1
one-shot-english	figure B.2
3-few-shot-english	figure B.3
3-few-shot-english-with-context	figure B.4
prompt-ner-english	figure B.5
one-shot-english-instruct*	figure C.4
3-few-shot-english-caps*	figure C.1
3-few-shot-english-with-context-instruct*	figure C.5
4-few-shot-english*	figure C.2
5-few-shot-english*	figure C.3

Table 3.3: Names and references of used prompts. The exact prompts can be found in the appendix. The last five prompts were only used for fine-tuning purposes.

3.3 Experimentation setup

A grid search was done to find the best performing combination of model, parameters and prompt template. The experimentation service was developed for calculating and performing the grid search from all given input (model, prompt template, parameters and test dataset). It connects to MLflow[4] to persist the results (see Figure 3.3). The results are also stored locally on disk of the experimentation service. The MLflow instance and the experimentation service both run in the on-premise compute cluster. MLflow stores its state on disk. This way no data has to leave the cluster as the cluster has no internet access.

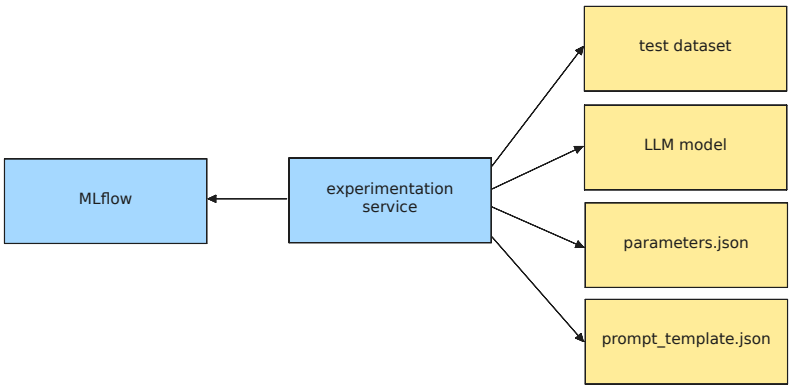

Figure 3.3: Components that were used for the experimentation. All experimentation results were sent to a locally running MLflow instance through the MLflow API.

The parameters are stored in a JSON file (see Figure 3.4). They can be either a fixed value or an array of values. The experimentation service uses them to create a matrix of all possible combinations of all the values. The prompt template is also a JSON file that is loaded into memory. The actual notes from the test dataset replace the placeholders in the template. The resulting prompt is then used for inference. The output of the inference is stored as part of the results. The results consist of the used parameters (one entry in the matrix), the expected labels, the predicted labels coming from the model inference, and the metrics. These are stored in a folder on disk (see Figure 3.5). The folder name is generated through hashing the used parameters with MD5. MLflow receives the input prompt, parameters used, inference output, and metrics. Having such a setup with the given automation ensures properly documented experiments. It provides traceability of which input lead to what output and performance at any point in time. After each inference step, the results are persisted. In case any experiment aborts or anything goes wrong, the temporary results are not lost. This also helps in development with fast feedback cycles.

This kind of setup also enabled to run multiple experiments in parallel and was robust enough against terminations ahead of successful completion. When a specific experiment terminated, it could just be restarted and the service can check which results were already calculated and which are missing. Worst case, it deleted an incomplete folder within the results folder of an experiment. It also made the experiments fully traceable, being able to properly track which inputs lead to what results.

[4] https://web.archive.org/web/20240126141203/https://mlflow.org/

```
{
  "max_length": [
    50,
    100
  ],
  "temperature": [
    0.0,
    0.9
  ],
  "context_length": 2048
}
```

Figure 3.4: An example of the parameters configuration file (`parameters.json`). It would create four parameter configurations: $[50, 0.0, 2048]$, $[50, 0.9, 2048]$, $[100, 0.0, 2048]$, $[100, 0.9, 2048]$.

```
./experiments/mistral/one-shot-english
├── results
│   ├── 1e966bf4e2e4a8e6e7090cbf02c15d11
│   │   ├── config.json
│   │   ├── labels.csv
│   │   ├── metrics.json
│   │   └── predicted.csv
│   └── 4b1777b5836eac129283302961120e15
│       ├── config.json
│       ├── labels.csv
│       ├── metrics.json
│       └── predicted.csv
├── parameters.json
└── prompt_template.json
```

Figure 3.5: An example of the experimentation file structure. This folder structure would be for running mistral with the one-shot-english prompt template. There are two results stored.

3.4 Evaluation of experiments

MLflow was used heavily in the evaluation of the experiments. It was used to get an overview of all data collected, compare experiment results and find the best parameters and models. This was partially done by the tracking functionality of MLflow[5]. Within the tracking concept of MLflow tracking, an LLM model name would map to a MLflow experiment. The MD5 hash of the models parameters (see Figure 3.5) was mapped to a MLflow run. The parameters were logged through MLflows "log_params" API. The metrics were tracked through MLflows "log_metrics" API after each inference evaluation. It enabled seeing at which points in time the overall evaluation improved or worsened. MLflow introduced an API only for LLMs with "llm.log_predicitions". As a result it was also possible to see which exact LLM inputs lead to which exact inference

[5] `https://web.archive.org/web/20240126161847/https://mlflow.org/docs/2.9.2/tracking.html#concepts`

outputs. Signaling MLflow a run as finished, also helped in keeping track of the time it took for an experiment to finish and comparing the runtime of each LLM.

The evaluation function used to evaluate the inference result of an LLM is token based.
Both the labels and the output of the LLM inference were pre-processed before the actual evaluation to get to tokens. The pre-processing is comprised of:

1. remove any occurrences of "Dr", "Prof", "med", "." and any double whitespaces
2. split the string by "," resulting in multiple substrings of firstname and lastname
3. split by whitespace resulting in tokens of names including firstnames and lastnames
4. trim the whitespace around the names to get to clean tokens
5. remove duplicates

This, for example, turned the string "Prof. Dr. med. Jane Doe, Dr. John Johnson" into $[" Jane", " Doe", " John", " Johnson"]$. The evaluation function would then compare the two lists of tokens from the expected labels and the results of the inference. It ignores the order in which the tokens appear. A correctly found token would count as true positive. A token that exists in the prediction but not in the actual label would be interpreted as a false positive. A token that was missing in the prediction would count as false negative. The evaluation function counts a predicted empty label and an empty actual label as true negative (see Table 3.4). With the sum of all true positives, false positives, true negatives and false negatives all standard evaluation metrics like accuracy, precision or recall can be calculated.

De-identification of notes poses a particular challenge in terms of false negatives, and therefore, the recall metric becomes crucial. As previously mentioned in the introduction, it is more detrimental if the model fails to identify an existing name (false negative) than if it incorrectly identifies a non-existent name (false positive). A low accuracy, on the other hand, can lead to the elimination of valuable information from the notes due to the presence of numerous false positives that are not names. Therefore, a balance of high accuracy and high recall is necessary for effective de-identification.

Actual tokens	**Predicted tokens**	**Metrics**
[""]	[""]	1 true negative
["Jane", "Doe"]	["John", "Doe"]	1 false positive, 1 false negative, 1 true positive

Table 3.4: Examples for token evaluation done inside the evaluation function.

Chapter 4

Results

4.1 German prompts

The German prompts were only tested with Llama-2. All of the exact results can be found in Appendix D. Table 4.1 gives an overview over the prompts and the parameters that lead to the highest accuracy and recall. The prompts can be found in Appendix A.

Prompt	Max. length	Temperature	Context length	Accuracy	Recall
zero-shot-german-with-context	50	0.5	2048	0.0298	0.7203
zero-shot-german-with-context	250	0.0	2048	0.022	0.9576
3-few-shot-german	50	0.9	2048	0.0424	0.9576
3-few-shot-german	50	0.0	1024	0.0333	1.0
3-few-shot-german-with-context	50	0.0	1024	0.1401	0.9831
3-few-shot-german-with-context	100	0.9	2048	0.1338	1.0
prompt-ner-german	50	0.9	2048	0.2185	0.0508
prompt-ner-german	250	0.5	2048	0.0438	0.7119

Table 4.1: Overview of German prompts (see Appendix A) and parameters that lead to the highest accuracy and recall for Llama-2. The detailed results can be found in Appendix D.

4.2 English prompts used with Llama-2

4.2.1 zero-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.0278	0.8898
50	0.5	1024	**0.0293**	0.8729
50	0.9	1024	0.0196	0.5763
50	0.0	2048	0.0278	0.8898
50	0.5	2048	0.0245	0.7627
50	0.9	2048	0.0217	0.6356
100	0.0	1024	0.0232	0.9153
100	0.5	1024	0.0213	0.8136
100	0.9	1024	0.0189	0.5847
100	0.0	2048	0.0232	0.9153
100	0.5	2048	0.0199	0.7627
100	0.9	2048	0.0183	0.5169
250	0.0	1024	0.0227	**0.9407**
250	0.5	1024	0.0177	0.6949
250	0.9	1024	0.0148	0.4492
250	0.0	2048	0.0227	0.9407
250	0.5	2048	0.021	0.7627
250	0.9	2048	0.0196	0.5678

Table 4.2: Results for the experiments running with Llama-2 and the zero-shot-english prompt. Figure B.1 shows the used prompt.

4.2.2 one-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.1206	0.9831
50	0.5	1024	0.128	0.9492
50	0.9	1024	**0.1323**	0.9407
50	0.0	2048	0.1206	0.9831
50	0.5	2048	0.1238	0.9661
50	0.9	2048	0.1196	0.9492
100	0.0	1024	0.1127	0.9831
100	0.5	1024	0.1244	**1.0**
100	0.9	1024	0.1178	0.9322
100	0.0	2048	0.1127	0.9831
100	0.5	2048	0.1087	0.9746
100	0.9	2048	0.1121	0.8983
250	0.0	1024	0.1117	0.9831
250	0.5	1024	0.1156	0.9746
250	0.9	1024	0.1116	0.9407
250	0.0	2048	0.1117	0.9831
250	0.5	2048	0.1131	0.9576
250	0.9	2048	0.1188	0.8898

Table 4.3: Results for the experiments running with Llama-2 and the one-shot-english prompt. Figure B.2 shows the used prompt.

4.2.3 3-few-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.1557	**0.9746**
50	0.5	1024	**0.1743**	0.9492
50	0.9	1024	0.1144	0.7373
50	0.0	2048	0.1557	0.9746
50	0.5	2048	0.1694	0.9512
50	0.9	2048	0.1223	0.8305
100	0.0	1024	0.1313	0.9746
100	0.5	1024	0.1342	0.9322
100	0.9	1024	0.1071	0.8051
100	0.0	2048	0.1313	0.9746
100	0.5	2048	0.1493	0.9661
100	0.9	2048	0.1043	0.7119
250	0.0	1024	0.1299	0.9746
250	0.5	1024	0.1324	0.9237
250	0.9	1024	0.1059	0.7797
250	0.0	2048	0.1299	0.9746
250	0.5	2048	0.1387	0.9153
250	0.9	2048	0.1089	0.7881

Table 4.4: Results for the experiments running with Llama-2 and the 3-few-shot-english prompt. Figure B.3 shows the used prompt.

4.2.4 3-few-shot-english-with-context

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.9381	0.6864
50	0.5	1024	0.8888	0.7203
50	0.9	1024	0.7083	0.7203
50	0.0	2048	0.9381	0.6864
50	0.5	2048	0.8778	0.6525
50	0.9	2048	0.6741	0.5593
100	0.0	1024	0.9381	0.6864
100	0.5	1024	0.8989	0.678
100	0.9	1024	0.7219	0.678
100	0.0	2048	0.9381	0.6864
100	0.5	2048	**0.9412**	**0.8136**
100	0.9	2048	0.7399	0.7797
250	0.0	1024	0.9381	0.6864
250	0.5	1024	0.911	0.7797
250	0.9	1024	0.7078	0.7542
250	0.0	2048	0.9381	0.6864
250	0.5	2048	0.9175	0.7034
250	0.9	2048	0.7109	0.6695

Table 4.5: Results for the experiments running with Llama-2 and the 3-few-shot-english-with-context prompt. Figure B.4 shows the used prompt.

4.2.5 prompt-ner-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.3209	0.0339
50	0.5	1024	**0.3419**	0.0085
50	0.9	1024	0.2828	0.0254
50	0.0	2048	0.3209	0.0339
50	0.5	2048	0.3156	0.0339
50	0.9	2048	0.269	0.0
100	0.0	1024	0.0915	0.3136
100	0.5	1024	0.1236	0.3898
100	0.9	1024	0.1801	0.3983
100	0.0	2048	0.0915	0.3136
100	0.5	2048	0.0996	0.2458
100	0.9	2048	0.1509	0.2627
250	0.0	1024	0.0628	0.3814
250	0.5	1024	0.0723	0.3983
250	0.9	1024	0.1147	0.3136
250	0.0	2048	0.0628	0.3814
250	0.5	2048	0.0735	**0.4746**
250	0.9	2048	0.1347	0.2966

Table 4.6: Results for the experiments running with Llama-2 and the prompt-ner-english prompt. Figure B.5 shows the used prompt.

4.3 English prompts used with Mistral

4.3.1 zero-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.0819	0.7542
50	0.5	1024	0.1112	0.7034
50	0.9	1024	**0.155**	0.7627
50	0.0	2048	0.0819	0.7542
50	0.5	2048	0.1211	0.6525
50	0.9	2048	0.1414	0.6525
100	0.0	1024	0.069	0.8051
100	0.5	1024	0.1125	0.8559
100	0.9	1024	0.1233	0.6949
100	0.0	2048	0.069	0.8051
100	0.5	2048	0.1035	0.7458
100	0.9	2048	0.1239	0.8051
250	0.0	1024	0.0691	**0.9237**
250	0.5	1024	0.0764	0.7373
250	0.0	2048	0.0691	0.9237
250	0.5	2048	0.0879	0.7241

Table 4.7: Results for the experiments running with Mistral and the zero-shot-english prompt. Figure B.1 shows the used prompt.

4.3.2 one-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.8672	**0.9831**
50	0.5	1024	0.8393	0.9746
50	0.9	1024	0.5265	0.9831
50	0.0	2048	0.8672	0.9831
50	0.5	2048	0.7553	0.9831
50	0.9	2048	0.5111	0.9746
100	0.0	1024	**0.8682**	0.9831
100	0.5	1024	0.7365	0.9831
100	0.9	1024	0.5247	0.9831
100	0.0	2048	0.8682	0.9831
100	0.5	2048	0.7673	0.9831
100	0.9	2048	0.4731	0.9661
250	0.0	1024	0.8682	0.9831
250	0.5	1024	0.6453	0.9831
250	0.9	1024	0.524	0.9831
250	0.0	2048	0.8682	0.9831
250	0.5	2048	0.6689	0.9831
250	0.9	2048	0.4862	0.9322

Table 4.8: Results for the experiments running with Mistral and the one-shot-english prompt. Figure B.2 shows the used prompt.

4.3.3 3-few-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	**0.7255**	0.7458
50	0.5	1024	0.6394	0.7542
50	0.9	1024	0.4742	0.8729
50	0.0	2048	0.7255	0.7458
50	0.5	2048	0.6429	0.822
50	0.9	2048	0.4752	0.7712
100	0.0	1024	0.6461	0.7458
100	0.5	1024	0.5549	**0.8983**
100	0.9	1024	0.4748	0.7458
100	0.0	2048	0.6461	0.7458
100	0.5	2048	0.593	0.8136
100	0.9	2048	0.4178	0.7797
250	0.0	1024	0.6368	0.7458
250	0.5	1024	0.6007	0.822
250	0.9	1024	0.466	0.822
250	0.0	2048	0.6368	0.7458
250	0.5	2048	0.5658	0.8983
250	0.9	2048	0.477	0.7712

Table 4.9: Results for the experiments running with Mistral and the 3-few-shot-english prompt. Figure B.3 shows the used prompt.

4.3.4 3-few-shot-english-with-context

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	**0.9948**	**0.9746**
50	0.5	1024	0.977	0.9576
50	0.9	1024	0.9706	0.9407
50	0.0	2048	0.9948	0.9746
50	0.5	2048	0.9672	0.9661
50	0.9	2048	0.9412	0.9407
100	0.0	1024	0.9948	0.9746
100	0.5	1024	0.982	0.9576
100	0.9	1024	0.9244	0.8814
100	0.0	2048	0.9948	0.9746
100	0.5	2048	0.9808	0.9746
100	0.9	2048	0.9309	0.9322
250	0.0	1024	0.9948	0.9746
250	0.5	1024	0.9795	0.9576
250	0.9	1024	0.9072	0.9237
250	0.0	2048	0.9948	0.9746
250	0.5	2048	0.9871	0.9576
250	0.9	2048	0.9177	0.9237

Table 4.10: Results for the experiments running with Mistral and the 3-few-shot-english-with-context prompt. Figure B.4 shows the used prompt.

4.3.5 prompt-ner-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	**0.8138**	0.0508
50	0.5	1024	0.6638	0.0508
50	0.9	1024	0.5742	0.0339
50	0.0	2048	0.8138	0.0508
50	0.5	2048	0.6829	0.0339
50	0.9	2048	0.6176	0.0169
100	0.0	1024	0.6291	0.4831
100	0.5	1024	0.547	0.4661
100	0.9	1024	0.3719	0.4237
100	0.0	2048	0.6291	0.4831
100	0.5	2048	0.4878	0.4576
100	0.9	2048	0.3867	0.3898
250	0.0	1024	0.5004	0.7797
250	0.5	1024	0.3399	0.7458
250	0.9	1024	0.2686	**0.8051**
250	0.0	2048	0.5004	0.7797
250	0.5	2048	0.3654	0.7712
250	0.9	2048	0.2722	0.7458

Table 4.11: Results for the experiments running with Mistral and the prompt-ner-english prompt. Figure B.5 shows the used prompt.

4.3.6 3-few-shot-english-caps

Max. length	Temperature	Context length	Accuracy	Recall
100	0.0	2048	0.9948	0.9746

Table 4.12: Fine-tuning result of using Mistral with the 3-few-shot-english-caps prompt (see Figure C.1). This experiment uses the parameters of the 3-few-shot-english-with-context prompt with the highest accuracy and recall. It was conducted to see if adding uppercase names improves the result compared to the 3-few-shot-english-with-context prompt.

4.3.7 4-few-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
100	0.0	2048	0.2963	0.9492

Table 4.13: Fine-tuning results of using Mistral with the 4-few-shot-english prompt (see Figure C.2). This experiment uses the parameters of the 3-few-shot-english-with-context prompt with the highest accuracy and recall. It was conducted to see if adding more examples improves the result compared to the 3-few-shot-english-with-context prompt.

4.3.8 5-few-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
100	0.0	2048	0.9922	0.9576

Table 4.14: Fine-tuning results of using Mistral with the 5-few-shot-english prompt (see Figure C.3). This experiment uses the parameters of the 3-few-shot-english-with-context prompt with the highest accuracy and recall. It was conducted to see if adding more examples improves the result compared to the 3-few-shot-english-with-context prompt.

4.4 English prompts used with Mistral instruct

4.4.1 one-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	**0.9**	0.9831
50	0.5	1024	0.8395	0.9831
50	0.9	1024	0.5882	0.9746
50	0.0	2048	0.9	0.9831
50	0.5	2048	0.7446	0.9831
50	0.9	2048	0.5447	0.9407
100	0.0	1024	0.9	0.9831
100	0.5	1024	0.8159	0.9831
100	0.9	1024	0.5182	0.9831
100	0.0	2048	0.9	0.9831
100	0.5	2048	0.7278	0.9831
100	0.9	2048	0.573	0.9661
250	0.0	1024	0.9	0.9831
250	0.5	1024	0.7479	0.9831
250	0.9	1024	0.514	0.9746
250	0.0	2048	0.9	0.9831
250	0.5	2048	0.8453	**0.9915**
250	0.9	2048	0.5673	0.9322

Table 4.15: Results for the experiments running with Mistral instruct and the one-shot-english prompt. Figure B.2 shows the used prompt.

4.4.2 one-shot-english-instruct

Max. length	Temperature	Context length	Accuracy	Recall
100	0.0	2048	**0.0298**	**0.9831**
100	0.5	2048	0.0281	0.9576
100	0.9	2048	0.0212	0.8983

Table 4.16: Results for the experiments running with Mistral instruct and the one-shot-english-instruct prompt. Figure C.4 shows the used prompt. The parameters with the highest accuracy and recall in Table 4.8 were taken with different values in temperature.

4.4.3 3-few-shot-english-with-context

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	**0.9948**	**0.9746**
50	0.5	1024	0.9922	0.9576
50	0.9	1024	0.9418	0.9492
50	0.0	2048	0.9948	0.9746
50	0.5	2048	0.9909	0.9492
50	0.9	2048	0.9657	0.9322
100	0.0	1024	0.9948	0.9746
100	0.5	1024	0.9897	0.9492
100	0.9	1024	0.9743	0.8898
100	0.0	2048	0.9948	0.9746
100	0.5	2048	0.9897	0.9661
100	0.9	2048	0.9708	0.9576
250	0.0	1024	0.9948	0.9746
250	0.5	1024	0.9948	0.9746
250	0.9	1024	0.9845	0.9576
250	0.0	2048	0.9948	0.9746
250	0.5	2048	0.9744	0.9322
250	0.9	2048	0.9501	0.9322

Table 4.17: Results for the experiments running with Mistral instruct and the 3-few-shot-english-with-context prompt. Figure B.4 shows the used prompt.

4.4.4 3-few-shot-english-with-context-instruct

Max. length	Temperature	Context length	Accuracy	Recall
100	0.0	2048	**0.8927**	**0.8983**
100	0.5	2048	0.2903	0.8136
100	0.9	2048	0.1025	0.6186

Table 4.18: Results for the experiments running with Mistral instruct and the 3-few-shot-english-with-context-instruct prompt. Figure C.5 shows the used prompt. The parameters with the highest accuracy and recall in Table 4.10 were taken with different values in temperature.

4.5 English prompts used with Falcon

4.5.1 zero-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.0293	0.7881
50	0.5	1024	0.0647	0.7627
50	0.9	1024	**0.1073**	0.6695
50	0.0	2048	0.0293	0.7881
50	0.5	2048	0.0723	**0.8051**
50	0.9	2048	0.099	0.6271
100	0.0	1024	0.027	0.8051
100	0.0	2048	0.0317	0.7471

Table 4.19: Results for the experiments running with Falcon and the zero-shot-english prompt. The experiments had to be stopped earlier because they took about 4 hours each. That is why there are not as many compared to the other. Figure B.1 shows the used prompt.

4.5.2 one-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.0519	**0.9915**
50	0.5	1024	0.0574	0.9915
50	0.9	1024	0.064	0.9492
50	0.0	2048	0.0519	0.9915
50	0.5	2048	0.0633	0.9831
50	0.9	2048	0.0687	0.9746
100	0.0	1024	0.0511	0.9915
100	0.5	1024	0.0574	0.9915
100	0.9	1024	0.0692	0.9831
100	0.0	2048	0.0511	0.9915
100	0.5	2048	0.0597	0.9915
100	0.9	2048	0.0632	0.9661
250	0.0	1024	0.051	0.9915
250	0.5	1024	0.055	0.9915
250	0.9	1024	**0.0756**	0.9576
250	0.0	2048	0.051	0.9915
250	0.5	2048	0.06	0.9831
250	0.9	2048	0.0725	0.9661

Table 4.20: Results for the experiments running with Falcon and the one-shot-english prompt. Figure B.2 shows the used prompt.

4.5.3 3-few-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.2367	**0.9661**
50	0.5	1024	0.2276	0.9661
50	0.9	1024	0.232	0.8898
50	0.0	2048	0.2367	0.9661
50	0.5	2048	0.2095	0.9492
50	0.9	2048	0.2153	0.9153
100	0.0	1024	**0.237**	0.9661
100	0.5	1024	0.2226	0.9407
100	0.9	1024	0.2188	0.8983
100	0.0	2048	0.237	0.9661
100	0.5	2048	0.2332	0.9492
100	0.9	2048	0.2303	0.9153
250	0.0	1024	0.2362	0.9661
250	0.5	1024	0.226	0.9661
250	0.9	1024	0.1928	0.8983
250	0.0	2048	0.2362	0.9661
250	0.5	2048	0.2143	0.9322
250	0.9	2048	0.1876	0.9492

Table 4.21: Results for the experiments running with Falcon and the 3-few-shot-english prompt. Figure B.3 shows the used prompt.

4.5.4 3-few-shot-english-with-context

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	**0.9832**	**0.8983**
50	0.5	1024	0.9338	0.7712
50	0.9	1024	0.8248	0.7627
50	0.0	2048	0.9832	0.8983
50	0.5	2048	0.9168	0.7458
50	0.9	2048	0.7897	0.7119
100	0.0	1024	0.9832	0.8983
100	0.5	1024	0.9085	0.7712
100	0.9	1024	0.7933	0.7203
100	0.0	2048	0.9832	0.8983
100	0.5	2048	0.9164	0.7712
100	0.9	2048	0.826	0.7458
250	0.0	1024	0.9832	0.8983
250	0.5	1024	0.9306	0.822
250	0.9	1024	0.7685	0.7797
250	0.0	2048	0.9832	0.8983
250	0.5	2048	0.9145	0.7542
250	0.9	2048	0.7872	0.678

Table 4.22: Results for the experiments running with Falcon and the 3-few-shot-english-with-context prompt. Figure B.4 shows the used prompt.

4.5.5 prompt-ner-english

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.7975	0.0169
50	0.5	1024	0.5211	0.0678
50	0.9	1024	0.0	0.0
50	0.0	2048	0.7975	0.0169
50	0.5	2048	0.5762	0.0932
50	0.9	2048	0.2772	0.1017
100	0.0	1024	0.0	0.0
100	0.5	1024	0.0	0.0
100	0.9	1024	0.0	0.0
100	0.0	2048	**0.8037**	0.1864
100	0.5	2048	0.0	0.0
100	0.9	2048	0.0	0.0
250	0.0	1024	0.0	0.0
250	0.5	1024	0.0	0.0
250	0.9	1024	0.0	0.0
250	0.0	2048	0.6635	**0.2797**
250	0.5	2048	0.0	0.0
250	0.9	2048	0.1648	0.2627

Table 4.23: Results for the experiments running with Falcon and the prompt-ner-english prompt. Figure B.5 shows the used prompt.

4.6 English fine tune prompts used with Falcon instruct

4.6.1 one-shot-english

Max. length	Temperature	Context length	Accuracy	Recall
100	0.0	2048	0.1047	**0.9831**
100	0.5	2048	0.1776	0.9831
100	0.9	2048	**0.24**	0.9407

Table 4.24: Results for the experiments running with Falcon instruct and the one-shot-english prompt. Figure B.2 shows the used prompt. The parameters with the highest accuracy and recall in Table 4.20 were taken with different values in temperature.

4.6.2 3-few-shot-english-with-context

Max. length	Temperature	Context length	Accuracy	Recall
100	0.0	2048	**0.9378**	0.6017
100	0.5	2048	0.9344	0.661
100	0.9	2048	0.9061	**0.6949**

Table 4.25: Results for the experiments running with Falcon instruct and the 3-few-shot-english-with-context prompt. Figure B.4 shows the used prompt. The parameters with the highest accuracy and recall in Table 4.22 were taken with different values in temperature.

4.7 Inference duration

Inference duration is simplified by taking the overall time it took to complete an experiment. This includes running the whole test dataset through the model and evaluate after each inference step. For each model type the median was taken. Only the experiments with the English prompts were used for this analysis. The instruct specified models and the the Mistral fine-tuning experiments were excluded from this, as they did not run through the same parameters.

Model	Median duration	Mean duration	Standard deviation	Min.	Max.
Llama-2	1.0h	1.9294h	2.7893	0.4h	24.0h
Mistral	0.8h	1.074h	0.8166	0.2h	3.0h
Falcon	0.5h	1.4892h	1.7616	0.2h	6.5h

Table 4.26: Statistical measures are shown for the duration of all the conducted experiments and the LLMs.

Table 4.26 shows the means and other statistical measures of the duration of the LLMs. Evaluating Mistral and Falcon with a Welch's t-test to determine if the mean duration of Mistral is smaller than the mean duration of Falcon, yields a p-value of 0.0242, which is smaller than $\alpha = 0.05$. Therefore, Mistral is the fastest model, given the hardware resources the compute cluster offered.

Chapter 5

Discussion

5.1 Performance comparison of English and German prompts

The results from Llama-2 show performance differences between German and English prompts (see Tables 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6). The differences are subtle but discernible. Especially the results for the 3-few-shot-german-with-context and the 3-few-shot-english-with-context prompt show drastically different accuracy values. The German prompts in general show very low accuracy and very high recall values. This could indicate, that the model does not understand what it should do and just tries to repeat the words it sees in the note.

Providing an English context and instruction within the prompt enhances the overall accuracy and recall values of Llama-2. This improvement can probably be attributed to the fact that Llama-2 was predominantly trained on datasets in English, although other languages were also included [Touvron et al. 2023b]. It could leave the impression of Llama-2 understanding better what to do with English instructions. Consequently, Llama-2 is capable of processing English prompts that contain German lab notes. The English prompt increases Llama-2's likelihood of identifying relevant patterns based on the instruction provided, thereby improving the accuracy and the recall of the result.

5.2 Performance comparison of LLMs

All employed models exhibit varying performances. The specific prompt and parameters that yield the highest accuracy and precision are utilized for performance comparison. According to the results, the top-performing LLM is Mistral, which significantly outperforms Falcon and Llama-2 in terms of recall (see Table 5.1). Although the difference in accuracy between Falcon and Mistral is not as substantial as the difference in recall, it still exists.

Model	Max. length	Temperature	Context length	Accuracy	Recall
Llama-2	100	0.5	2048	0.9412	0.8136
Mistral	50	0.0	1024	**0.9948**	**0.9746**
Falcon	50	0.0	1024	0.9832	0.8983

Table 5.1: Best performing results of all models on the 3-few-shot-english-with-context prompt. See Tables 4.5, 4.10, 4.22 for more details.

Mistral outperforming Llama-2 is consistent with the observations made in [Jiang et al. 2023]. However, Falcon 7B outperforming Llama-2 7B is something not completely in line with the results from [Almazrouei et al. 2023]. This behavior suggests that the results of the benchmarks and datasets used do not apply to German medical context. Isolating exactly what architectural

difference attributes to this performance difference in this particular case is rather complex. One thing that could influence this is the difference in the datasets that were used in training both models. Both models use the transformer architecture and have made adjustments to it primarily for efficiency and performance reasons. These adjustments are in place so that the models can with given time and resources train over more data. It is rather likely that the difference in their training data plays a larger factor in the performance difference. This makes a lot of sense if you consider the core functionality and architecture of a transformer. Falcon prominently uses the RefinedWeb dataset [Penedo et al. 2023]. One could only speculate about how well suited e.g. the RedinedWeb dataset is for the specific task of identifying people and their names from German medical notes. This observation extends to the entirety of the datasets used for training both models.

5.3 Zero-shot, one-shot or few shot prompts?

All models show similar performance differences between the used prompts. The only prompt that produced acceptable results was the 3-few-shot-english-with-context prompt. This prompt worked best for all the models. All other prompts showed massive problems with accuracy rather than recall. All the LLMs had difficulties producing accurate results with less than three examples given and instead had lots of false positives leading to low accuracy. Most of the time they even identified information that was in the text as an individual's name. This leads to information loss inside the then de-identified note.

The zero-shot prompts showed the lowest performance across all LLMs. Most of the time the output of the LLMs did not make sense at all and seemed to just repeat certain words that were in the note. Adding just one example, leading to the one-shot prompt, did improve the accuracy but is still far away from the performance of the 3-few-shot-english-with-context prompt. It seems that solely from the context sentence piece plus instruction, the LLMs cannot deduce what to do semantically. They need examples to know what to look for. The outputs of the LLMs showed also that having a negative example of a sentence without a person, helped improve the overall performance. The context sentence in its own is also impacting the performance positively (see results for 3-few-shot-english in comparison to 3-few-shot-english-with-context). Upon examining the number of examples required, it is evident that having more than three examples does not improve but decrease performance (see Table 4.13 and 4.14). Mistral did produce a lot of false positives for the 4-few-shot-english prompt, that were mostly numbers. It further worsened the accuracy. This could be due to the examples of the prompt. One example had multiple numbers. It may indicate that the model struggles with German texts and assumes that numbers are more likely to appear since it did not see enough training data in German to establish patterns for what a person's name looks like. This is especially visible, since the 5-few-shot-english prompt did not contain this specific example.

Using Mistral instruct together with the instruct optimized prompts performed worse than the normal Mistral with the 3-few-shot-english-with-context prompt. The outputs of the 3-few-shot-english-with-context-instruct prompt show a hint on to why this is the case. The outputs show extra tokens that indicate some sort of structure to the inference result. Some examples of the output: "* Dr. Jane Doe", "* None", ""‘\nJohn Doe \n“‘". This is likely due to the finetuning on the instruct specialized dataset. That dataset has lots of structured data outputs. The outputs of the 3-few-shot-english-with-context prompt and the 3-few-shot-english-with-context-instruct prompt unfortunately both showed almost the same false negatives.

The primary factor contributing to the performance of the output of the prompts, as mentioned earlier, is that LLMs do not necessarily learn abstract concepts like a "person" or a "name of a

person". They are based on transformers, which are constructed using an encoder and a decoder. The decoder part of a transformer, responsible for generating output, produces tokens that are likely to appear next based on a prior input sequence. It learns this probability from its training datasets. This goes more towards repeating what other people are telling you and less towards learning abstract concepts and having true knowledge of what a person is.

5.4 The 1.0 recall in Llama-2

In Figure 4.3, Figure 4.1 and the figures in Appendix D some recall values are 1.0. It means that all the names were successfully recognized. However, the accuracy and precision are very low, suggesting that the overall number of false positives is also very high. Llama-2 did produce for those prompts lots of "names" of people, which are random words from the notes. Some are numbers, units, or just any other word that appeared in the note in combination with words that did not appear in the note. It suggests a missing understanding from the prompts of what to do.

Additionally, Figure 4.3 shows a temperature of 0.5, which also produces hallucinations. The high temperature value increases the randomness of the model's output, making it more prone to generating unexpected or inaccurate responses. Consequently, all the experiments of Llama-2 with a recall of 1.0 can be considered a fluke rather than a reliable outcome.

5.5 PromptNER performance

All experiments with the PromptNER prompt over all models did not outperform the 3-few-shot-english-with-context prompt. The prompt-ner-english prompt experiments showed vastly better accuracy and precision scores compared to the zero-shot-english prompt. The one-shot-english prompt did outperform the prompt-ner-english prompt though. Increasing the max. length of the created output of the LLM does favor the result. This is due to the fact that the chain of thought format needs more tokens in the output. The high accuracy can be attributed to the chain of thought reasoning the prompt generates. The low recall shows the missing understanding of a person and their name though. The outputs show lots of examples of false labeling and reasoning about a name. Sometimes a name is a noun, sometimes an adjective, which is leading to false negatives. Sometimes the reasoning is complete nonsense, especially when the temperature is not 0.0, leading to hallucinations. Utilizing few-shot prompts appears to be much more effective, not only in terms of accuracy, precision, and recall, but also in terms of inference-time. The few-shot prompts did run through the LLMs faster than the PromptNER prompts. This is due to the high amount of tokens the chain of thought format produces. Most few-shot prompt experiments took around 20-30 minutes to finish. The PromptNER prompt experiments took around 2-6 hours to finish.

Falcon specifically shows significant problems with the prompt-ner-english prompt, leading to lots of results where the only output it produced was ";;". This is why accuracy and recall metrics are 0.0 for many of the experiments (see Table 4.23). This could again be the result of the datasets that were used to train Falcon, since it is not directly tied to the temperature, the context length or the maximum answer length.

5.6 Impact of context length and maximum answer length

Throughout all experiments, the context length of 1024 and 2048 exhibit very similar performance patterns. All experiment results with a temperature of 0.0, the same max. length, but

a changing context length between 1024 and 2048 are further highlighted for this. The results show no differences in the recall and accuracy for a context length of 1024 or 2048. This leads to the conclusion that the context length in these experiments did not contribute to any significant influence over the results. This is very likely considering Mistral with a predefined context length of 8192 and Falcon and Llama-2 with a predefined context length of 4096. Restricting the context length did not have any effect.

5.7 Impact of temperature on performance

Temperature is a parameter in the softmax function:

$$P(y_i) = \frac{e^{z_i/\tau}}{\sum_j e^{z_j/\tau}}, \text{ where} \tag{5.1}$$

$P(y_i)$ is the probability of the i-th output (softmax function output)
z_i is the raw score (logit) associated with the i-th output
τ is the temperature

Increasing the temperature causes the softmax function activation to generate more evenly distributed $P(y_i)$ probabilities. It is therefore adding more noise to the probabilities, leading to more entropy. This in turn means other pathways inside the neural network are so to say explored.
The results show, it can improve the accuracy (see Table 4.19) or the precision (see Table 4.9) or both at the same time (see Table 4.5). For the specific context of de-identifying medical lab notes, the temperature is a very dangerous parameter. You would ideally expect the model to consistently identify a person's name accurately. However, introducing a sufficient degree of entropy into the model may result in instances where the model randomly fails to recognize a person's name, leading to false negatives and rendering the entire dataset unusable. Although further fine-tuning might enhance overall metrics, the inherent risk to the model's reliability for production use in this task remains a concern.

5.8 Overall quality of results

It is noteworthy that 7B LLMs demonstrate the ability to generate results with accuracy, precision and recall values within the 90% range. This suggests that the model size is sufficient to perform named entity recognition on people in medical lab notes to a certain degree.

Mistral has produced in its best performance three false negatives. One of the false positives is coming from the note piece:

> "[...] Abnahme: Verfahren von Mitarbeiter DOE an ABC Ambulanz (1337) [...]"

The name in that note is "DOE".
The prompt 3-few-shot-english-caps was crafted with one example having a name written in capital letters (see Figure C.1). This was done to assess the feasibility of mitigation. The results of that experiment show no difference and the false negative is still present (see Table 4.12). This is likely due to the fact of Mistral not understanding that a "Mitarbeiter" indicates a person.
Another false negative is of the form:

> "[...] MRSA (kulturell): nicht nachgewiesen Befundverantwortlich Dr. Doe [...]"

Here the LLM should be able to identify "Dr. Doe" as a person based on the "Dr." title. The lack of German medical context dataset in the training of Mistral could lead to Mistral not being able

to generalize from "Befundverantwortlich" that a person is meant by that.

Another interesting example is the only false positive that stems from the note piece:

> "[...] Wert telefonisch durchgegeben am: 04.02.23 um: 12:07 von: Jane Doe an:133742 Grenzwert [...]"

The false positive that was detected is "133742". In the context of the lab note a person with the name of "Jane Doe" is letting another person know information via phone call. The other person is indicated through "an:133742". It is true, that this is not a specific name of a person but it is still an identifier representing a person. In the general context you could argue that this is a person, but that is not the name of the person.

5.9 Special observations made during development and experimentation

At the outset of this thesis, the field of large language models (LLMs) has been rapidly advancing, with almost daily improvements in various aspects of the technology. It started with the introduction of running inference with C, moved over to LoRa, quantization and to faster and smaller models. At one point, the quantized model format was switched from ggml to gguf, which subsequently increased overall performance. This made the development challenging as it had to be agile at a rapid scale. This was compensated by using a three-tier architecture[1] and dependency injection through higher level functions and functional programming. Models and libraries were easily abstracted away through these techniques and the internal domain logic was kept intact.

Before the evaluation that is mentioned above was used, an evaluation function based on exact string matching was used. This did not work well as the LLMs output were often slightly different than the expected labels. The expected labels were manually created with the titles of the names (e.g. Prof., Dr., med.) and cleansed of any double whitespaces or tabs. The LLMs were often skipping the titles or sometimes split the names between firstname and lastname or ordered the names in a different order than the label. This then led to the evaluation function interpreting it as a failure although the actual names were there. The modular nature of the code mentioned before helped here again in just replacing the evaluation function with the more robust one from the section above.
During the experimentation a major error in the calculation of the now token based evaluation was noticed. Because of the structure in Figure 3.5 the evaluation function could just rerun on the results labels.csv and predicted.csv files. The experiments did not have to be aborted and started over from scratch.

The compute cluster environment had built in timeouts for processes. The experiments themselves took different times. It was necessary to keep some of them running for multiple days. This timeout behavior was limiting this. The compute cluster environment would send SIGHUP signals to the processes after the timeout, terminating them. This was overcome through the nohup command. It runs a process and ignores all incoming SIGHUP signals.

Various leaderboards on Hugging Face also helped tracking the overall development state of

[1] `https://web.archive.org/web/20231215081031/https://martinfowler.com/bliki/PresentationDomainDataLayering.html`

new and existing models and their performance[2]. The thesis first started out with just Llama, then MedAlpaca was released and shortly after Falcon, Llama-2 and after that Mistral. During development, MedAlpaca was abandoned due to factors such as the lack of a released gguf model. Most models are taken from Hugging Face from the user "TheBloke"[3]. The latest models could be found there at reasonable delay after release as a quantized gguf model with good documentation.

[2] https://web.archive.org/web/20240125005033/https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

[3] https://web.archive.org/web/20240114041017/https://huggingface.co/TheBloke

Chapter 6

Conclusion

The experiments and their results show that named entity recognition can be done on CPU with Mistral 7B and how it outperforms Llama-2 7B and Falcon 7B. Quantized models are usable in CPU only environments. When prompting for the specific task of de-identifying people through recognizing their names in medical lab notes, a prompt with context, an instruction and 3 examples is the best way to do it, comparing it to other n-shot (including zero-shot and one-shot) prompts. The length of the output of the LLM only impacts the performance if the output can be higher than the configured length. A length of 50 is supposedly enough in combination with the 3-few-shot-english-with-context prompt. A context length lower than 8192 does not have any impact on the output of Mistral 7B for this specific setting.

Furthermore, the results show that using Mistral in production for automating this task is rather dangerous and impractical. Firstly, there are false negatives which appeared even in the best combination of parameters and prompt. This makes it unusable in terms of data privacy. Secondly, using an LLM that takes for 700 rows in a dataset a median of 48 minutes is too slow, especially when dealing with big data. It is more likely that Mistral 7B can be used for this purpose as a additional system next to, for instance, a deterministic solution. One can imagine also a regex based solution together with Mistral to lead to a practical solution. Another realistic setting could be a database with all the names of employees to be used as a cross reference and then the LLM to additionally double check the result from that database query.
Even building and training a custom neural net for this specific task could potentially lead to better results (or using it in a hybrid system together with an LLM). Using an LLM in clinical contexts as usability help and second safety net when dealing with manual entry data seems like the most logical use case for now.

The practical examples just highlighted show already how wasteful it can be to use an LLM for this purpose. A regex uses drastically less resources to identify a name compared to the LLM. Mistral 7B even in its quantized form takes a significant time to run inferences, indicating a huge electricity usage and therefore CO2 footprint by comparison. To justify this, the overall usage scenario needs to be carefully evaluated in terms of alternatives and their complexity, resource constraints and overall investment.

Another aspect the results show is the knowledge LLMs really hold. LLMs are based on transformers, trained on a high amount of large datasets of texts. They learn probabilities of tokens to appear based on previously seen tokens. They do not really learn abstract concepts of knowledge like what a name of a person is. On top of that Falcon, Llama-2 and Mistral show significant problems with German texts and medical texts, as they hold very specific vocabulary, that is most likely not present in their training data. The simple act of putting "Mitarbeiter" in front of a name and writing it in all capital letters will lead to Mistral 7B not recognizing this name. In one case

Mistral was not even able to identify a word with the title "Dr." in front of it is a name of a person.

Nevertheless, Mistral is able to recognize names to a degree. Further mitigation strategies to improve the performance could be more specific adjustments to the used prompt. Adding a hint in the prompt like "If you can not find a person, denote that as 'None'." could improve the overall performance. Giving it more specific context like "People in this context can be staff in hospitals, doctors or patients." could also lead to it being able to identify the false negatives it was not able to. Another possible improvement could be using the LLM or something else to translate the note into English first and then identifying the people mentioned in it. Fine-tuning Mistral on German medical context data taken from real use cases, can also lead to a better LLM similar to MedAlpaca.
For this thesis the given dataset was only analyzed to a certain degree. Doing more analysis like for example word counts resulting in word clouds, leading to better data quality or embeddings that can be used in combination with Mistral, would also be good ideas to further boost the performance.

At the current state Mistral should not be used for the specific reasons and limitations mentioned here. Nevertheless, the data that was used in its own shows high potential in advancing LLMs and current research, as the models struggled to perfectly execute this seemingly trivial task. The diversity in datasets that are used to train LLMs has to improve in order for them to get anywhere close to artificial general intelligence in terms of texts. The value of an LLM in clinical contexts is still at this current stage already high. It could be used for helping people encode ICD-10, detect anomalies in medical documentation of a patient or help people make decisions when determining their examinations or diagnoses. The important part here is that LLMs at this point in time should only be used to assist people. They are not good enough in clinical contexts yet to automatically derive decisions but to only assist people in their decision making process.

Appendix A

Exact German prompts used

Du bist ein Experte in Linguistik und extrahierst Namen von Personen aus Texten. Der Name kann aus einen Nachnamen und einem Vornamen bestehen. Namen können zusätzlich Titel oder eine Anrede beinhalten. Extrahiere aus dem unten stehenden Satz die Namen von Personen.

Satz: '{sentence}'
Personen:

Figure A.1: Simple German zero-shot prompt with context.

Q: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
A: Dr. med. Paola Mertens, Dr. med. Peter Worms
###
Q: Wert telefonisch durchgegeben am: 22.07.23 um: 13:37 von: Petra Rams an: Tom Grenzwert (99. Perzentil) zum Ausschluss myokardialer Ischämie: 11 ng
A: Petra Rams, Tom
###
Q: 421338 Ct-Wert Target 1: Diensthabender Arzt : Dr. med. Hans-Peter Lobeck
A: Dr. med. Hans-Peter Lobeck
###
Q: {sentence}
A:

Figure A.2: German few-shot prompt with three examples without context.

Du bist ein Experte in Linguistik und extrahierst Namen von Personen aus Texten. Der Name kann aus Kombinationen aus einem Nachnamen, einem Vornamen, einem Titel oder einer Anrede bestehen. Es folgen Namen von Personen in den unten stehenden Beispielen.

Beispiel: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
Namen von Personen: Dr. med. Paola Mertens, Dr. med. Peter Worms
###
Beispiel: Wert telefonisch durchgegeben am: 22.07.23 um: 13:37 von: Petra Rams an: Tom Grenzwert (99. Perzentil) zum Ausschluss myokardialer Ischämie: 11 ng
Namen von Personen: Petra Rams, Tom
###
Beispiel: 421338 Ct-Wert Target 1: Diensthabender Arzt : Dr. med. Hans-Peter Lobeck
Namen von Personen: Dr. med. Hans-Peter Lobeck
###
Beispiel: {sentence}
Namen von Personen:

Figure A.3: German few-shot prompt with three examples and with context.

Defn: Eine Entität ist eine Person. Eine Person hat einen Namen, der aus einem optionalen Titel oder einer Andrede, einem Vornamen und einem Nachnamen bestehen kann. Der Name einer Person kann aus mehreren Wörtern und Satzzeichen bestehen. Datum, Zeiten, Adjektive, Verben, Zahlen, Zahlenzeichen oder Kombinationen aus alle diesen sind keine Person.

Q: Identifiziere alle möglichen Entitäten im unten stehenden Paragraphen. Begründe für jeden Eintrag, warum es sich hier um eine Entität handelt oder warum nicht.
###
Paragaph: "421337 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"

Antwort:
1. 421337 | False | weil es eine Zahl ist und somit keine Person
2. Auftragsnummer | False | weil es ein einfaches Nomen ist und kein Name, also keine Person
3. Diensthabender Arzt | False | weil es ein Nomen mit einem adjektiv ist, also keine Person
4. Dr. med. Paola Mertens | True | weil das eine diensthabende Ärztin ist, also eine Person
5. Dr. med. Peter Worms | True | weil das ein diensthabender Arzt ist, also eine Person
###
Paragraph: {sentence}

Antwort:

Figure A.4: German PromptNER prompt. It contains of a definition, a question and task and an example note and answer in a chain of thought format.

Appendix B

Exact English prompts used

You are an expert in linguistics and extract people with names from texts. A name can consist of a firstname and a lastname or of multiple of these. Names can also consist of titles. Extract from the following sentence the people.
Sentence: '{sentence}'
People:

Figure B.1: Simple English zero-shot prompt with context.

You are an expert in linguistics and extract people with names from texts. A name can consist of a firstname and a lastname or of multiple of these. Names can also consist of titles. Extract from the following sentence the people.
###
Sentence: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
People: Paola Mertens, Peter Worms
###
Sentence: '{sentence}'
People:

Figure B.2: English one-shot prompt with context. This is the zero-shot prompt but with one example.

Q: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
A: Paola Mertens, Peter Worms
###
Q: Wert telefonisch durchgegeben am: 22.07.23 um: 13:37 von: Petra Rams an: Tom Grenzwert (99. Perzentil) zum Ausschluss myokardialer Ischämie: 11 ng
A: Petra Rams, Tom

###
Q: Anwendung: 1-0-0-0 Stk. vom 13.01.23 bis 18.01.23Stk.
A: None
###
Q: {sentence}
A:

Figure B.3: English few-shot prompt with three examples of which one does not contain a name.

You are an expert in linguistics and extract people with names from texts. A name can consist of a firstname and a lastname or of multiple of these. Names can also consist of titles. Extract from the following sentence the people.
###
Sentence: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
People: Paola Mertens, Peter Worms
###
Sentence: Wert telefonisch durchgegeben am: 22.07.23 um: 13:37 von: Petra Rams an: Tom Grenzwert (99. Perzentil) zum Ausschluss myokardialer Ischämie: 11 ng
People: Petra Rams, Tom
###
Sentence: Anwendung: 1-0-0-0 Stk. vom 13.01.23 bis 18.01.23Stk.
People: None
###
Sentence: '{sentence}'
People:

Figure B.4: English few-shot prompt with three examples of which one does not contain a name and with context.

Defn: A person has a name. A name can consist of multiple words and titles. A name does not contain numbers. Abstract scientific concepts are not a person even if they have a name associated with them. Dates, times, adjectives and verbs are not a person.

Q: Given the paragraph below, identify a list of possible persons and for each entry explain why it either is or is not a person:
###
Paragaph: "421337 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"

Answer:
1. 421337 externe Auftragsnummer | False | as this contains a number (421337)
2. Diensthabender Arzt | False | as this is an adjective (Diensthabender) and a noun (Arzt)
3. Dr. med. Paola Mertens | True | as this is a person with a name that follows the general format of a name
4. Dr. med. Peter Worms | True | as this is a person with a name that follows the general format of a name
###
Paragraph: {sentence}

Answer:

Figure B.5: English PromptNER prompt. It contains of a definition, a question and task and an example note and answer in a chain of thought format.

Appendix C

Exact prompts used for fine tuning

You are an expert in linguistics and extract people with names from texts. A name can consist of a firstname and a lastname or of multiple of these. Names can also consist of titles. Extract from the following sentence the people.
###
Sentence: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
People: Paola Mertens, Peter Worms
###
Sentence: Wert telefonisch durchgegeben am: 22.07.23 um: 13:37 von: Petra Rams an: Mitarbeiter TOMARI Grenzwert (99. Perzentil) zum Ausschluss myokardialer Ischämie: 11 ng
People: Petra Rams, TOMARI
###
Sentence: Anwendung: 1-0-0-0 Stk. vom 13.01.23 bis 18.01.23Stk.
People: None
###
Sentence: '{sentence}'
People:

Figure C.1: Few-shot prompt with 3 examples, containing one without a name and one with a name written in all capital letters.

You are an expert in linguistics and extract people with names from texts. A name can consist of a firstname and a lastname or of multiple of these. Names can also consist of titles. Extract from the following sentence the people.
Sentence: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
People: Paola Mertens, Peter Worms
###
Sentence: Wert telefonisch durchgegeben am: 22.07.23 um: 13:37 von: Petra Rams an: Tom Grenzwert (99. Perzentil) zum Ausschluss myokardialer Ischämie: 11 ng People: Petra Rams, Tom
###

Sentence: Anwendung: 1-0-0-0 Stk. vom 13.01.23 bis 18.01.23Stk.
People: None
###
1550978366 Hinweise z. Abnahme: Telefonische Durchsage erfolgte am 10.03.2022 um 12:32 mittels Readback- 1550978366 Hinweise z. Abnahme: Verfahren von Mitarbeiter KRÜMPNERI an HNO Ambulanz (2194) 1550978366 Hinweise z. Abnahme: Abstrich unbeschriftet
People: KRÜMPNERI
###
Sentence: '{sentence}'
People:

Figure C.2: Few-shot prompt with 4 examples, containing one without a name and one with a name written in all capital letters.

You are an expert in linguistics and extract people with names from texts. A name can consist of a firstname and a lastname or of multiple of these. Names can also consist of titles. Extract from the following sentence the people.
###
Sentence: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
People: Paola Mertens, Peter Worms
###
Sentence: Wert telefonisch durchgegeben am: 22.07.23 um: 13:37 von: Petra Rams an: Tom Grenzwert (99. Perzentil) zum Ausschluss myokardialer Ischämie: 11 ng
People: Petra Rams, Tom
###
Sentence: Anwendung: 1-0-0-0 Stk. vom 13.01.23 bis 18.01.23Stk.
People: None
###
Sentence: "*** Endbefund *** vom 02.09.2022 09:22, validiert von Dr. Matthias Grossu am 02.09.2022"
People: Matthias Grossu
###
Sentence: "796738 Ct-Wert Target 1: Diensthabender Arzt : Prof. Dr. med. Petra Sucheisen, Dr. med. Karola Bedo"
People: Petra Sucheisen, Karola Bedo
###
Sentence: '{sentence}'
People:

Figure C.3: Few-shot prompt with 5 examples, containing one without a name.

<s>[INST] You are an expert in linguistics and extract people with names from texts. A name can consist of a firstname and a lastname or of multiple of these. Names can also consist of titles. Extract from the following sentence the people.
Sentence: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
People:[/INST] Paola Mertens, Peter Worms
###</s>
[INST] Sentence: '{sentence}'
People:[/INST]

Figure C.4: One-shot prompt with Mistral instruct optimized prompt.

<s>[INST] You are an expert in linguistics and extract people with names from texts. A name can consist of a firstname and a lastname or of multiple of these. Names can also consist of titles. Extract from the following sentence the people.
Sentence: "133742 externe Auftragsnummer: Diensthabender Arzt : Dr. med. Paola Mertens, Dr. med. Peter Worms"
People:[/INST] Paola Mertens, Peter Worms
###
[INST] Sentence: Wert telefonisch durchgegeben am: 22.07.23 um: 13:37 von: Petra Rams an: Tom Grenzwert (99. Perzentil) zum Ausschluss myokardialer Ischämie: 11 ng
People:[/INST] Petra Rams, Tom
###
[INST] Sentence: Anwendung: 1-0-0-0 Stk. vom 13.01.23 bis 18.01.23Stk.
People:[/INST] None
###</s>
[INST] Sentence: 'sentence'
People:[/INST]

Figure C.5: Few-shot prompt with 3 examples, containing one without a name. This prompt is instruct optimized for Mistral.

Appendix D

Detailed experiment results for the German prompts

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.0279	0.9407
50	0.5	1024	0.027	0.7119
50	0.9	1024	0.0273	0.4576
50	0.0	2048	0.0279	0.9407
50	0.5	2048	**0.0298**	0.7203
50	0.9	2048	0.0257	0.4322
100	0.0	1024	0.023	0.9492
100	0.5	1024	0.0206	0.5847
100	0.0	2048	0.023	0.9492
100	0.5	2048	0.0228	0.6864
250	0.0	2048	0.022	**0.9576**

Table D.1: Results for the experiments running with Llama-2 and the zero-shot-german-with-context prompt. This prompt took around 4h or more to finish for one experiment, which is why there are less experiments. Figure A.1 shows the used prompt.

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.0333	**1.0**
50	0.5	1024	0.0361	0.9915
50	0.9	1024	0.0397	0.9492
50	0.0	2048	0.0333	1.0
50	0.5	2048	0.0358	0.9915
50	0.9	2048	**0.0424**	0.9576
100	0.0	1024	0.0293	1.0
100	0.5	1024	0.0314	0.9831
100	0.9	1024	0.0355	0.9661
100	0.0	2048	0.0293	1.0
100	0.5	2048	0.0315	0.9746
100	0.9	2048	0.038	0.9407
250	0.0	1024	0.0289	1.0
250	0.5	1024	0.0288	0.9746
250	0.9	1024	0.037	0.9915
250	0.0	2048	0.0289	1.0
250	0.5	2048	0.0306	0.9915
250	0.9	2048	0.0372	0.9661

Table D.2: Results for the experiments running with Llama-2 and the 3-few-shot-german prompt. Figure A.2 shows the used prompt.

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	**0.1401**	0.9831
50	0.5	1024	0.1319	0.9831
50	0.9	1024	0.1342	0.9831
50	0.0	2048	0.1401	0.9831
50	0.5	2048	0.1366	0.9915
50	0.9	2048	0.1331	0.9915
100	0.0	1024	0.1273	0.9831
100	0.5	1024	0.1278	0.9915
100	0.9	1024	0.1207	0.9915
100	0.0	2048	0.1273	0.9831
100	0.5	2048	0.1155	0.9915
100	0.9	2048	0.1338	**1.0**
250	0.0	1024	0.1255	0.9831
250	0.5	1024	0.1164	0.9915
250	0.9	1024	0.1151	0.9915
250	0.0	2048	0.1255	0.9831
250	0.5	2048	0.1143	0.9915
250	0.9	2048	0.1328	0.9831

Table D.3: Results for the experiments running with Llama-2 and the 3-few-shot-german-with-context prompt. Figure A.3 shows the used prompt.

Max. length	Temperature	Context length	Accuracy	Recall
50	0.0	1024	0.084	0.0254
50	0.5	1024	0.1349	0.0254
50	0.9	1024	0.2006	0.0339
50	0.0	2048	0.084	0.0254
50	0.5	2048	0.1544	0.0254
50	0.9	2048	**0.2185**	0.0508
250	0.0	2048	0.0269	0.6356
250	0.5	2048	0.0438	**0.7119**

Table D.4: Results for the experiments running with Llama-2 and the prompt-ner-german prompt. This prompt took around 4h to finish for one experiment, which is why there are less experiments. Figure A.4 shows the used prompt.

Bibliography

[bfa 2023] Icd-10-gm international statistical classification of diseases, german modification. https://www.bfarm.de/EN/Code-systems/Classifications/ICD/ICD-10-GM/_node.html. [Accessed 29-12-2023].

[hel 2023] Über das Unternehmen Helios: Wer wir sind — helios-gesundheit.de. https://www.helios-gesundheit.de/unternehmen/wer-wir-sind/. [Accessed 01-12-2023].

[Abu-El-Rub et al. 2022] Abu-El-Rub, N., Urbain, J., Kowalski, G., Osinski, K., Spaniol, R., Liu, M., Taylor, B., und Waitman, L. R., Natural language processing for enterprise-scale de-identification of protected health information in clinical notes. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2022:92–101.

[Ahmed et al. 2021] Ahmed, A., Abbasi, A., und Eickhoff, C., Benchmarking modern named entity recognition techniques for free-text health record de-identification.

[Almazrouei et al. 2023] Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Étienne Goffinet, Hesslow, D., Launay, J., Malartic, Q., Mazzotta, D., Noune, B., Pannier, B., und Penedo, G., The falcon series of open language models.

[Ashok und Lipton 2023] Ashok, D. und Lipton, Z. C., Promptner: Prompting for named entity recognition.

[Brown et al. 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., und Amodei, D., Language models are few-shot learners.

[Grundmann et al. 2021] Grundmann, P., Arnold, S., und Löser, A., Self-supervised answer retrieval on clinical notes.

[Han et al. 2023] Han, T., Adams, L. C., Papaioannou, J.-M., Grundmann, P., Oberhauser, T., Löser, A., Truhn, D., und Bressem, K. K., Medalpaca – an open-source collection of medical conversational ai models and training data.

[He et al. 2018] He, Y.-L., Zhang, X.-L., Ao, W., und Huang, J. Z., Determining the optimal temperature parameter for softmax function in reinforcement learning. *Applied Soft Computing*, 70:80–85.

[Hu et al. 2021] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., und Chen, W., Lora: Low-rank adaptation of large language models.

[Jiang et al. 2023] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., und Sayed, W. E., Mistral 7b.

[Lample et al. 2016] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., und Dyer, C., Neural architectures for named entity recognition.

[Lester et al. 2021] Lester, B., Al-Rfou, R., und Constant, N., The power of scale for parameter-efficient prompt tuning. Seiten 3045–3059.

[Liu et al. 2022] Liu, X., Chen, H., und Xia, W., Overview of named entity recognition. *Journal of Contemporary Educational Research*, 6(5):65–68.

[Penedo et al. 2023] Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., und Launay, J., The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

[Richter-Pechanski et al. 2019] Richter-Pechanski, P., Amr, A., Katus, H. A., und Dieterich, C., Deep learning approaches outperform conventional strategies in de-identification of german medical reports. *Studies in health technology and informatics*, 267:101–109.

[Roberts et al. 2021] Roberts, M., Driggs, D., Thorpe, M., Gilbey, J., Yeung, M., Ursprung, S., Aviles-Rivero, A. I., Etmann, C., McCague, C., Beer, L., Weir-McCall, J. R., Teng, Z., Gkrania-Klotsas, E., Ruggiero, A., Korhonen, A., Jefferson, E., Ako, E., Langs, G., Gozaliasl, G., Yang, G., Prosch, H., Preller, J., Stanczuk, J., Tang, J., Hofmanninger, J., Babar, J., Sánchez, L. E., Thillai, M., Gonzalez, P. M., Teare, P., Zhu, X., Patel, M., Cafolla, C., Azadbakht, H., Jacob, J., Lowe, J., Zhang, K., Bradley, K., Wassin, M., Holzer, M., Ji, K., Ortet, M. D., Ai, T., Walton, N., Lio, P., Stranks, S., Shadbahr, T., Lin, W., Zha, Y., Niu, Z., Rudd, J. H. F., Sala, E., und Schönlieb, C.-B., Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nature Machine Intelligence*, 3(3):199–217.

[Sanh et al. 2021] Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N., Datta, D., Chang, J., Jiang, M. T.-J., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Fevry, T., Fries, J. A., Teehan, R., Bers, T., Biderman, S., Gao, L., Wolf, T., und Rush, A. M., Multitask prompted training enables zero-shot task generalization.

[Seuss et al. 2017] Seuss, H., Dankerl, P., Ihle, M., Grandjean, A., Hammon, R., Kaestle, N., Fasching, P., Maier, C., Christoph, J., Sedlmayr, M., Uder, M., Cavallaro, A., und Hammon, M., Semi-automated de-identification of german content sensitive reports for big data analytics. *RöFo - Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren*, 189:661–671.

[Shen et al. 2023] Shen, H., Chang, H., Dong, B., Luo, Y., und Meng, H., Efficient llm inference on cpus.

[Touvron et al. 2023a] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., und Lample, G., Llama: Open and efficient foundation language models.

[Touvron et al. 2023b] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B.,

Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., und Scialom, T., Llama 2: Open foundation and fine-tuned chat models.

[Tseo et al. 2020] Tseo, Y., Kumar, A., Salkola, M. I., Freddy, M. A., und Mohamed, A., Information extraction of clinical trial eligibility criteria.

[Vajjala und Balasubramaniam 2022] Vajjala, S. und Balasubramaniam, R., What do we really know about state of the art ner?

[van Aken et al. 2022] van Aken, B., Papaioannou, J.-M., Naik, M. G., Eleftheriadis, G., Nejdl, W., Gers, F. A., und Löser, A., This patient looks like that patient: Prototypical networks for interpretable diagnosis prediction from clinical text.

[Wang et al. 2018] Wang, C., Cho, K., und Kiela, D., Code-switched named entity recognition with embedding attention.

[Wynants et al. 2020] Wynants, L., Van Calster, B., Collins, G. S., Riley, R. D., Heinze, G., Schuit, E., Albu, E., Arshi, B., Bellou, V., Bonten, M. M. J., Dahly, D. L., Damen, J. A., Debray, T. P. A., de Jong, V. M. T., De Vos, M., Dhiman, P., Ensor, J., Gao, S., Haller, M. C., Harhay, M. O., Henckaerts, L., Heus, P., Hoogland, J., Hudda, M., Jenniskens, K., Kammer, M., Kreuzberger, N., Lohmann, A., Levis, B., Luijken, K., Ma, J., Martin, G. P., McLernon, D. J., Navarro, C. L. A., Reitsma, J. B., Sergeant, J. C., Shi, C., Skoetz, N., Smits, L. J. M., Snell, K. I. E., Sperrin, M., Spijker, R., Steyerberg, E. W., Takada, T., Tzoulaki, I., van Kuijk, S. M. J., van Bussel, B. C. T., van der Horst, I. C. C., Reeve, K., van Royen, F. S., Verbakel, J. Y., Wallisch, C., Wilkinson, J., Wolff, R., Hooft, L., Moons, K. G. M., und van Smeden, M., Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. *BMJ*, Seite m1328.

[Yamada et al. 2020] Yamada, I., Asai, A., Shindo, H., Takeda, H., und Matsumoto, Y., Luke: Deep contextualized entity representations with entity-aware self-attention. Seiten 6442–6454. Association for Computational Linguistics.

[Zhang et al. 2023] Zhang, C., D'Haro, L. F., Tang, C., Shi, K., Tang, G., und Li, H., xdial-eval: A multilingual open-domain dialogue evaluation benchmark.