Project
Studiengang Master Machine Learning & Data Analytics

Patrick Müller

Customization of Large Language Models to User-Defined Data

Prüfer:	Prof. Dr. Ulrich Klauck
Einreichungsdatum	Februar 2024

Contents

List of Figures

1 Introduction

In the rapidly developing field of artificial intelligence, Large Language Models (LLMs) have emerged as a powerful tool for understanding and generating human-like text. This study aims to investigate the customization and fine-tuning of a question-answering LLM on text containing module descriptions and the content of university lectures from the Department of Computer Science at Aalen University. Leading LLM providers use very large models. OpenAI's ChatGPT-3 has 175 billion trainable parameters and requires a large and expensive infrastructure to train and deploy (Ala20). Thus, this work focuses on models that can run efficiently on consumer hardware. The model used for this work is the Mistral LLM with 7 billion parameters. The aim is to customize and adapt these smaller LLMs so that it can not only understand the context and content of these lectures but also answer questions about the study modules and subjects. This involves a thorough analysis of available LLM solutions, understanding their strengths and weaknesses, model fine-tuning as well as evaluating the model outputs. Furthermore, a crucial part of this study is the data - its source, structure, and relevance to the task at hand. A detailed description of the data used is provided, along with the steps taken to pre-process the data. This study will provide insights into the capabilities of smaller LLMs, their potential applications in education, and the challenges that arise during the process of dataset creation and fine-tuning. This knowledge could pave the way for more advanced, interactive and personalised chatbots and improved learning experiences in the future, where optimized LLMs can be used on more accessible infrastructure.

Natural language processing (NLP) is used to solve problems such as text classification (e.g. sentiment analysis), machine translation (e.g. spoken language translation) and the optimization of search engines and recommendation systems from e.g. shopping platforms (GZ21). NLP is concerned with how to design systems to process and evaluate large amounts of natural language data. Subtopics are natural language understanding (NLU) and natural language generation (NLG). The data is mostly unstructured and can be accessed in large quantities via the Internet. Text analytics can be used to extract valuable information from natural language, i.e. to improve business profitability (ANS23). NLP has a long history of research. NLP approaches were used in the Georgetown experiment in 1954. IBM and Georgetown University translated over 60 Russian sentences into English using hand-coded language rules. In the 1970s, hard-coded rules and grammars were developed to parse language. In the late 1980s, statistical models were introduced to the field, providing more robust NLP models. Today, deep learning techniques are one of the main approaches (Jon17). More advanced algorithms are being developed due to the increase in available text on the internet and increasing computational resources. In the context of LLMs, the model size, the dataset size and the amount of computing power used for

training are the most important factors affecting model performance. The number of model parameters N is usually in the billions for SOTA models. The size of the dataset D is usually measured in tokens. The amount of computation C describes the resources invested in model training. Today, computing requirements are measured in petaflops days: $10^{15} \times 24 \times 3600 = 8.64 \times 1019$. Increasing the computation shows a decrease in test loss. The loss decreases as a simple function of N, D or C following a power law relationship, i.e. scaling one of these factors will result in a proportional relative change in the other two and the model loss will decrease accordingly. Looking at the current approach, larger models perform better than smaller ones due to the number of parameters and the size of the dataset, as well as the computational investment (KMH+20). Therefore, the ongoing research involves the optimization of LLMs with fewer parameters. Another important concept is the encoding of natural language into tokens. This involves breaking down whole texts into smaller pieces, such as individual words. The length of sentences or datasets containing text is measured by the number of tokens to describe the size. A dataset on which GPT-3 was trained contained around 400 billion tokens. A model's context window, which describes the maximum amount of text a model can compute in an inference, is also measured in tokens (BMR+20). A common tokenization method for LLMs is byte-pair encoding, which was also used in GPT-2 & GPT-3 (RNSS18, BMR+20). To compute natural language on digital computers, the tokens have to be converted into so-called embeddings. Embeddings represent the meaning of a word in a vector space. The similarity of words is represented by the distance of in the vector space, so more similar words are closer together (MES19). There are different algorithms and approaches to convert natural text into embeddings, such as the Word2Vec approach, which uses shallow neural networks to generate embeddings from large texts (MCCD13). Another approach is called ELMo which models complex characteristics of words, such as syntax and semantics, and also distinguishes between the linguistic context in which words are written (PNI+18). The rest of the paper is structured as follows: First, the history of LLMs is summed up briefly. Then the main technologies used in LLMs are explained in the chapter 3. The following chapter discusses approaches to fine-tune an LLM. Chapter 5 explains the data collected & used for this work, followed by chapter 6 showing the implementation details such as training and hyperparameters. Finally, the results are evaluated in 7 and the thesis concludes with a summary and discussion in chapter 8.

2 A brief history of LLMs

With the release of ChatGPT, LLMs were made available to consumers, bringing the technology into the mainstream. The term GPT stands for generative pre-trained transformers, which are explained in the section 3.2. Interestingly the technology behind ChatGPT is not so new, the used transformer technology emerged in 2017 from Google Research (VSP+23). Since then, LLM research has become increasingly popular and organizations have increased their investment in the technology. For example, Microsoft, which has had a partnership with OpenAi since 2019, has extended its investment to the order of billions of dollars by 2023 (Mic23a). Google is also actively researching and investing into LLMs. In 2019 they introduced BERT, a transformer which however is not capable of generating new content (DCLT19). Furthermore, they released LaMDA in 2021 which is similar to GPT-3. Google hesitated to release LLM consumer applications, knowing that the models were not ready to adhere to Google's AI principles (EC21). However, under pressure to keep up with the latest developments, Google announced its most advanced LLM, called Gemini, in late 2023 and released it to the public in early 2024, replacing its previous flagship LLM, called BARD (Goo23). While OpenAI and Google never open-sourced their more advanced models, many other institutions and companies did. In February 2023, Meta released its open-source model Llama (TLI+23), and just 6 months later, its predecessor Llama2. The models range from 7 billion to 70 billion parameters (TMS+23). Microsoft released their little LLM called Phy (Mic23b). There have been other open-source releases such as StableLM by StabilityAI or Falcon by TIIUAE, Mistral by Mistral AI and many others. The released models tend to be between 3 and 70 billion parameters in size and are often optimized for use on cheap consumer hardware (Fac23). This made self-hosted LLMs accessible to a wider user base. In 2023, the publication of LLM-related papers increased sharply. Many researchers who had never worked in the NLP domain began to publish LLM papers. In addition, research became interdisciplinary, reaching many different fields such as law, medicine and education. Microsoft & Google produced the most LLM papers in 2023 (MBP+23). New approaches get proposed regularly like for example using several smaller sub-models in combination. In this approach, the input gets mapped to the most fitting sub-model. These models promise faster performance than single, larger models. One example is Mixtral 8x7B, which consists of eight models with 7 billion parameters. According to the authors, it outperforms Llama 2 70B with 6x faster inference (tea23). Furthermore, new approaches that address the limitations of the transformer architecture are introduced. For example, Mamba, a state space model that provides faster inference than transformers (GD23). As the research field of LLMs is very active, new papers are constantly being published, which requires a lot of attention and effort from researchers to keep up with new developments. With the widespread availability of LLMs, the fear of such models has also grown. Problems such as hallucinations, where the models simply cre-

ate information that does not exist, made people question whether they could trust such algorithms. Studies have shown that GPT-4 can propagate and amplify harmful societal biases when used as clinical decision support (ZLS+24). The CEO and founder of OpenAI, Sam Altman, testified before the US Senate that governments should regulate artificial intelligence to avoid significant harm to the world. The tools could disrupt the job market like never before and increase the spread of disinformation and bias. Altman supported the creation of a federal agency that could grant licences to create AI models above a certain threshold of capability (Zor23). However, there is widespread debate among researchers as to whether the current state of AI technology really is a thread, or whether it is simply not clearly understood by the majority of people. Yann LeCun, chief AI scientist at Meta, argues that current technology is still very limited and can't compete with human-level intelligence. He argues that the models don't really understand the world, that they don't have common sense, and that they need huge amounts of data to reach a level of intelligence that is not that great (Per23).

3 The Technology behind LLMs

LLMs are neural network-based systems that can process and generate natural language text at scale. They are trained on large amounts of text data, often from diverse sources and domains, and learn to capture the statistical patterns and semantic relationships within language. LLMs have shown remarkable performance in a variety of natural language processing (NLP) tasks, such as text summarization, machine translation, question answering, and text generation. This chapter explains the technology behind LLMs, focusing on the key components and architectures that enable them to handle complex and diverse languages. Firstly, the transformer architecture is discussed, followed by the generative pre-trained transformers and the technical challenges and limitations.

3.1 The Transformer

The following section is largely taken from (VSP+23), the paper that introduces the transformer architecture. Recurrent Neural Networks (RNNs), LSTMs, and Gated Recurrent Neural Networks have long been the most widely used approaches to sequence modelling and problems such as language modeling and machine translation. Recurrent models generate a sequence of hidden states from previous hidden states. In other words, they use recursion to generate new outputs based on previous outputs. This sequential process does not allow the computation to be parallelized, which makes training very memory-intensive, especially for longer sequences. The Transformer architecture replaces recursion with an attention mechanism that draws global dependencies between input and output. This allows more parallelization during training and leads to better model performance. The Transformer was the first model to rely entirely on self-attention to compute representations of inputs and outputs. Self-attention is a mechanism that relates different positions of a single sequence to compute a representation of the given sequence. It significantly outperforms other sentence embedding models. It does this by computing a 2D matrix of a sentence, where each word is compared to every other word for interdependence. In other words, it learns the context of a word by looking at the other words in the sentence (LFdS+17). The transformer model is based on the encoder-decoder structure. The encoder maps an input sequence $(x_1, ..., x_n)$ to a latent space $z = (z_1, ..., z_n)$. Based on z, the decoder then generates an output sequence $(y_1, ..., y_m)$. For the output, the model is autoregressive, i.e. it generates the next element based on all previously generated elements. The initial architecture is shown in figure 3.1. In the encoder, each layer Nx consists of two sub-layers, a multi-head self-attention mechanism and a simple feed-forward network. Each sub-layer has a residual connection followed by layer normalization. The encoder produces outputs in the dimension x

defined by the parameter $d_{model} = x$ The decoder has N layers with an additional third sub-layer performing multi-head attention over the output encoder stack. Again, residual links and layer normalization are used in each sub-layer. Unlike the encoder, a masked multi-head attention sub-layer is used to ensure that the predictions can only depend on the already known previous outputs. Each of the layers also contains a fully connected feedforward network. The proposed scaled dot product attention uses the weight matrices queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$ with dimension d_k and values V with dimension d_v. $\boldsymbol{Q}$ represents the current element that the model wants to focus on. $\boldsymbol{K}$ represents the elements that the model can attend to. V represents the information the model can extract from the attended elements. The output matrix is calculated as:

$$\boldsymbol{Attention}(\boldsymbol{Q}, \boldsymbol{K}, V) = \boldsymbol{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})V \tag{3.1}$$

This is identical to the dot product attention except for the introduced scaling factor of $\frac{1}{\sqrt{d_k}}$. The scaling is done to avoid extremely small gradients when the dot products become large. In addition, multi-head attention is introduced, where the queries, keys and values are linearly projected h times with different linear projections to d_q, d_k, d_v. Multiple attention mechanisms are running in parallel, essentially allowing the network to extract more information from the input than with a single attention head. This is similar to the use of multiple kernels in convolutional neural networks (CNNs) to produce feature maps with multiple output channels. This has the advantage that each head can focus on different parts of the input. On each projection, the attention function is computed in parallel, resulting in a d_v-dimensional output value. These final values are then concatenated and projected again to produce the final values. The authors show the computations in the formula 3.2 as:

$$\begin{aligned} \boldsymbol{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, V) &= \boldsymbol{Concat}(head_1, ..., head_h)\boldsymbol{W}^O \\ where head_i &= \boldsymbol{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, V\boldsymbol{W}_i^V) \end{aligned} \tag{3.2}$$

$\boldsymbol{W}_i^Q, \boldsymbol{W}_i^K, \boldsymbol{W}_i^V$ are parameter matrices (projections). The linear projections reduce the dimension by $\frac{d_{model}}{h}$ where h is the number of attention heads and d_{model} the output size of the attention heads. The authors use multi-head attention in three different ways. Firstly, the so-called encoder-decoder attention is used when generating new output sequences. The query matrix comes from the previous decoder layer, and the key and value matrices come from the output of the encoder. This allows the decoder to focus on the relevant parts of the input sequence when generating the output sequence. Secondly, self-attention layers are used in the encoder, where all keys, values and queries come from the previous layers of the encoder. Each position in the encoder can attend to all positions in the previous layer of the encoder, so the order of the words does not matter. The same is true for the decoder with the exception that only up to the current position, all previous positions can attend to all the other positions (masked multi-head attention). This means that the decoder only pays attention to words before the current one. In addition to the attention mechanism, each sublayer also contains a fully connected feed-forward network consisting of two linear transfor-

mations with a ReLU activation in between. Input and output tokens are transformed by the learned embeddings into vectors of dimension d_{model}. The learned linear transformation and the softmax function are used to convert the decoder output into predicted next-token probabilities. Positional encodings are used to inject information about the position of the tokens in the input sequence. In the transformer architecture, sinusoids are used to obtain the required positional information. In summary, the transformer architecture offers huge advantages in computational speed and is very efficient compared to other methods such as RNN or CNN, especially for longer sequences.

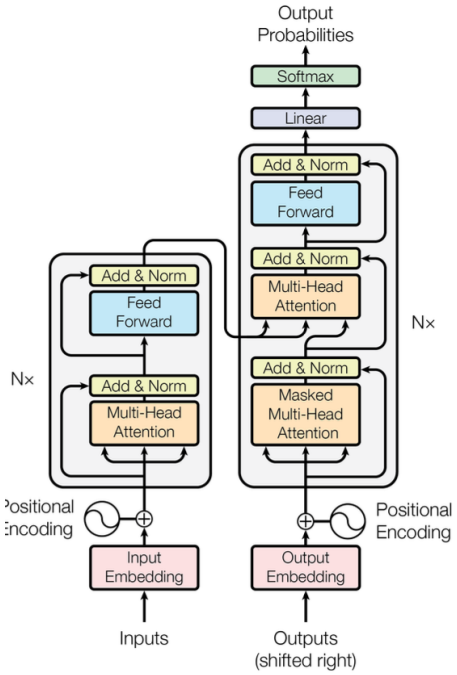

Figure 3.1: The transformer model architecture (VSP+23).

3.2 Generative pre-trained-transformers

GPTs are the most prominent type of LLM, also used in the famous ChatGPT model family (RN18). At the most basic level, they model a conditional distribution for an output sequence given an input sequence. The basic GPTs are trained in two steps. In the first step, a massive unsupervised text U is trained with the goal of maximizing the probability for a specific token u_i given a context window k of tokens that occur before u_i. Thus, the probability for a given token u_i is maximised by $L_1(U) = \sum_i \log P(u_i|u_{i-k}, ..., u_{i-1}; \Theta)$, where Θ represents the parameters of a neural network. The second step is supervised fine-tuning, where the parameters are adapted to a labeled data set C. The goal is to maximise the probability: $L_2(C) = \sum_{(x,y)} \log P(y|x^1, ..., x^m)$, where y is the label and x^m are the input tokens. Finally, an auxiliary objective is used that combines L_1 and L_2 as: $L_3(C) = L_2(C) + \lambda \times L_1(C)$ (RNSS18). With the introduction of GPT-3, it was shown that a massive increase in model parameters, dataset size and diversity, and train-

ing time makes increasingly efficient use of the unsupervised learning step. This increases the generative abilities, and allows GPTs like GPT-3 to be able to perform few-shot learning, where the model is given a few demonstrations of a task to perform, but no model weights are allowed to be updated, or even to perform zero-shot generation, where no demonstration example is provided and the model is given only a natural language instruction (BMR+20).

3.3 Common challenges and limitations

Transformers have problems with very long sequences. Approaches such as the previously mentioned Mamba, which use linear-time sequence modeling and selective states, promise to overcome this limitation (GD23). Furthermore, there are high computational requirements for large models. Training and inference require large amounts of memory, which most consumer hardware doesn't yet have. For example, GPT-3 requires 1.2TB of VRAM for training and 350GB of VRAM for inference. Inference is also very slow on non-optimised hardware, such as laptops that do not have a dedicated GPU. Techniques such as Low-Rank Adaptation (LoRA) & Weight-Decomposed Low-Rank Adaptation (DoRA) attempt to alleviate the memory problem by parameter-efficient fine-tuning (PEFT), which freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architectures (HSW+21, LWY+24). Figure 3.2 shows the reparametrization used in LoRA, where the newly introduced matrices A and B are combined with the already trained weight matrix W. The rank r determines the rank, where a lower rank like one results in fewer parameters. In frameworks like Lit-GPT it can be specified which projection matrices should be trained, e.g. the query, key, value or output projection matrices of an LLM. DoRA improves the learning capacity and stability of LoRA and consistently outperforms LoRA methods. DoRA decomposes the pre-trained weight into two components, magnitude and direction, for fine-tuning. It then uses LoRA to update only the directional component with a low-rank matrix, while keeping the magnitude component and the original weights fixed. There are also other approaches to fine-tuning LLMs, such as adapters (ZHL+23), where additional parameters are added to the transformer layers, which are outside the scope of this paper. Another way of achieving significant performance gains and speeding up inference is quantization. Most SOTA models use FP16 precision to store weights, which means they use 2 bytes to store a single weight. As parameter size increases, this leads to massive computational requirements. FP16 networks can be quantized to 8-bit or even 4-bit with little loss in performance. Neural networks have been shown to be robust to quantization, i.e. they can be quantized to lower bit widths with relatively little impact on the accuracy of the network. By quantizing weights and activations with low-bit integers, GPU memory requirements can be reduced. Weights stored in FP16 can then be quantized to 4-bit NormalFloat (DPHZ23, NFA+21). Lit-GPT, the framework used in this paper, uses the bitesandbytes library for quantization, which was introduced with the QLoRA approach, a method that combines LoRA with quantization. This approach makes it possible to fine-tune models on non-high-end hardware with SOTA results (DPHZ23). Another proposed solution is RAG (Retrieval Augmented Generation), introduced by Meta researchers in 2021. It

allows LLMs to be updated with data they have never seen before, without costly retraining or fine-tuning. This is very useful when the data needed by an LLM is not static and changes regularly. RAG retrieves data relevant to the query from a database that contains embedding vectors (LPP+21). LLMs are often limited by their context length. This is the maximum number of tokens the model can process in a single input. GPT-3 has a maximum context length of 2048 tokens (KHM+23, BMR+20). However, models such as Google Gemini can process up to 700,000 words at a time, overcoming the problem of short context length. Gemini 1.5 pro can be accessed with a context window of 128,000 tokens (Goo24). Another limitation is the need for huge pre-training datasets, collected through massive web scraping. Because of their size, quality assessments are difficult to perform, and the data can therefore contain a lot of unwanted content. LLMs also have a tendency to hallucinate. This means that LLMs make up facts and statements that are not accurate. For example, the models may refer to publications that do not exist. These hallucinations can be difficult to detect due to the fluent texts that LLMs produce (KHM+23). However, usage of RAG can reduce hallucinations of LLMs (LPP+21). One study found that pre-trained LLMs can degenerate into toxic text even from seemingly innocuous prompts. They also found uncensored toxic and biased documents in the GPT-2 training corpus that originated from banned websites or unreliable news sites (GGS+20). Jailbreaking can lead to security risks and exploits. A recent study shows that LLaMA-2 7B is vulnerable to adversarial attacks using programming instructions and language switching, as well as novel strategies such as role hacking and glitch tokens. In this way, security mechanisms implemented to make the model saver (GSS+24) can be bypassed. Possible improvements include the implementation of thorough fact-checking technologies to ensure that the content generated by the model is accurate and reliable. Future AI systems could use modification and control tools that allow users to shape the behavior of the AI according to their preferences. In addition, models should be trained with domain-specific knowledge for accurate task completion and expert assistance (KDBY23).

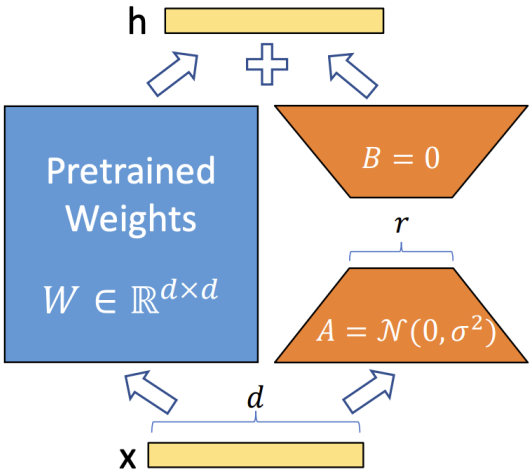

Figure 3.2: The LoRA approach (HSW+21).

4 Customization of an LLM with user-defined data

This chapter gives an overview of the current state of the art in adapting LLMs to user-defined data. As mentioned in section 3.3, training all layers on an LLM requires large computational resources. To facilitate customization, commercial LLM providers such as OpenAI offer consumer applications to mitigate this problem, discussed in section 4.1. There is also increasing research into optimizing smaller LLMs with PEFT approaches such as DoRA, discussed in section 4.2.

4.1 Commercial products

In November 2023, OpenAI announced the latest addition to its LLM applications called GPTs. GPTs allow users to create custom versions of ChatGPT for specific purposes. Users can create GPTs without coding and share them with others. They can perform various tasks such as web searching, image creation or data analysis. They also introduced a GPT store where users can find and use GPTs created by verified builders. Developers can connect GPTs to external APIs and data sources, and enterprise customers can deploy internal-only GPTs for their business needs. As of February 2024 the tool is only accessible for ChatGPT plus, team and enterprise users (Ope23). Using GPTs has the advantage that custom models can be created without investing in expensive hardware or expertise. However, the user must sacrifice privacy and control over the data, which can be particularly important for business-critical information. In addition, users of GPTs have reported that the application has limitations, e.g. GPTs cannot fully replace the intuition and nuanced understanding that comes from years of experience in a particular area (Cha24). Due to the massive cost of creating foundational LLMs on the scale of ChatGPT LLM LLM-based solutions like GPTs are currently only feasible for large organizations like OpenAI, Google, or Meta. Thus it's likely that solutions similar to GPTs will follow from companies like Meta or Google. Of course, there are other non-code solutions for creating chatbots that are not based on LLMs which are not discussed in this work.

4.2 Open-source solutions

Based on the Open LLM Score, the best performing open-source LLMs in February 2024 are Mixtral 2x11B with two 11 billion parameter models, followed by Mistral 7B

and LLaMa2 with 34 billion parameters (IM23). Mixtral & Mistral are both models from the French company Mistral AI. Given the frequency of improvements, this is likely to change soon. Mixtral models are so-called Sparse Mixture of Experts models that increase the number of parameters of a model while controlling cost and latency, as the model only uses a fraction of the total set of parameters per token. This is done by only using distinct layers (experts) in the interference. Thus, a Mixtral model with fewer parameters is able to outperform even models such as Llama 2 70B and GPT-3.5 (JSR+24). There exist a wide range of open-source models with small parameter counts like the already mentioned Microsoft phi-2 model with 2.7 billion parameters (Mic23b), to models like Falcon with up to 180 billion parameters (AAA+23). There are also several frameworks for easily training and fine-tuning existing open-source models. One of these is Lit-GPT (AI23), which was also used for fine-tuning in this work. Lit-GPT provides easy integration of open-source models and comes with pre-defined classes for easy and efficient workflows with LLMs. A further widely used platform is LangChain. LangChain provides tools and frameworks for simple and efficient development of applications using LLMs (Lan). Ollama is an application designed to run LLMs locally on consumer hardware. It is available for MacOS, Linux and Windows. It is popular for its ease of use and wide model support (Oll). Another popular platform for open-source LLMs is Hugging Face. They offer the most available models with useful implementations, not only for LLMs but for all kinds of models, intending to democratize good machine learning (Fac). General advantages of using open-source LLMs are that they are transparent and do not depend on the goodwill of a company. In addition, the use of open-source LLMs is essential for reproducibility in scientific research, which can be difficult when work is based on closed-source models (Spi23). Organizations face the challenge of integrating LLM technologies into their workflows. Open-source offers transparency, local deployment and privacy. They are also not bound by a vendor's terms of use and values. However, there is a need for domain knowledge if low-code or no-code solutions are not used. (A.R23). Open-source LLMs could be adapted like today's popular open-source programming languages (Spi23), however, it is hard to say whether closed-source models like ChatGPT will dominate in the future or open-source models will be the preferred solution.

5 Used data

Data is the foundation of any data science project and plays a crucial role in determining the quality and applicability of the results. This section describes how the data was collected, how the dataset was generated, and the issues and challenges encountered along the way.

5.1 Description

The data are unstructured text data from the website of the University of Aalen. On the website, all study programmes of the Faculty of Computer Science are presented. Study contents, staff, future perspectives and events are presented in order to attract new students or provide information for enrolled students. This information is mostly found directly in text on the website or in linked PDF documents. Images or other data such as audio were not included. To facilitate the information and decision-making process, all information should be provided to the user by the proposed LLM application. This makes the process of finding the right study course easier and more natural but is also useful for already enrolled students who need information about e.g. courses and exams.

5.2 Data acquisition and scraping

The BeautifulSoup (Sou) and Requests (Rei) packages were used for scraping. The aim was to get as much data as possible to cover all relevant information about the courses. Each course has an individual site directory with subdirectories such as downloads or staff working at the course. For each course, all directories and subdirectories were scraped. This resulted in a total of 439 .txt files from 12 study programs (3.1 megabytes of text). The text is mostly unstructured data, with the exception that some scraped tables could be saved in .csv files.

5.3 Data processing and labeling

The proposed dataset was structured based on the LIMA dataset (ZLX+23). LIMA was designed to train a helpful AI assistant based on question-answer pairs collected from various forums. Thus, it fits the proposed goal of this work to generate a question-answering LLM. The structure is shown in Figure 5.1, where each data

point is in a JSON Lines object. Each object contains a "conversations" array with a question and an answer. It also documents the source, which in this case is the study program from which the information was extracted. The ChatGPT API was used to create the dataset. This simplified the process as there was no need for the author to pre-process the data. Instead, the unstructured data could be fed directly into the API where ChatGPT automatically extracted questions and answers. The instructions given to the ChatGPT API are shown in Figure 5.2. Other open-source tools were tested, such as the Question Generator Pipeline form Haystack (Hay23), which allows various LLM models from Hugging Face to be inserted. However, the proposed models did not give satisfactory results, and for improvements larger models had to be tested, which would be very computationally and time-consuming. The model used for labeling was gpt-3.5-turbo-1106 with temperature=0.2 and top_p=0.1. The temperature and top_p variables both control the creativity of the model. Because the extracted answers & questions should be precise and informal, lower values were used. The reason for not using GPT-4 was that the cost would be on average 30 times higher using GPT-4, whereas GPT-3.5 costs only $0.0005 per 1000 input tokens and $0.0015 per 1000 output tokens. The final cost, including testing and generating the dataset, was $1.85. The ChatGPT dataset generation also showed some minor bugs, see figure 5.3. For example, the formatting of the JSON line document was sometimes incorrect and had to be corrected manually. The tagging produced redundant questions, i.e. the same or similar question and answer were repeated several times from the same document. Some question-answer pairs were incomplete, i.e. the answer was missing or only partially generated. To ensure that the content of the question-answer pairs was correct, the author took several samples from the dataset and checked them for correctness. In the case of an error, it was removed or corrected. Compared to other datasets, the generated dataset has a short sequence length. As seen in figure 5.4 the maximum sequence length is 294 tokens, with a mean of 36.5 & a median of 72.5 tokens. Compared to other datasets the sequences are not long. For example, the alpaca dataset (TGZ+23) has a median length of 110 and a maximum sequence length of 1304 tokens. However, the advantage of short sequence lengths is that training does not require as much computation as for datasets with very long sequences. ChatGPT was given no instructions on how long the sequences should be, but because the temperature was set relatively low (0.2) the outputs were not as random as with a higher temperature, which could also have reduced the overall sequence length. Given the goal of training an LLM to answer questions, sequence length is not important as long as the questions are answered with sufficient accuracy. For evaluating purposes a little dataset with additional 49 prompts was created. The dataset contained additional questions like variations that are not present in the train or test dataset or questions in German. The answers of the evaluation dataset were created by ChatGPT and reviewed by the author. The answers are more detailed than the answers in the training and test dataset.

```
{"conversations": ["What is the address of Hochschule Aalen?", "The
   address of Hochschule Aalen is Anton-Huber-Str. 25, 73430
   Aalen."], "source":"it_security_Bachelor"}
```

Figure 5.1: Structure of the dataset, each datapoint is stored in a JSON Lines object.

```
 messages=[
{"role": "system",
"content": "You are an assistant who extracts questions and the
   corresponding answers from texts to create a dataset for a
   machine learning model, you return only valid .jsonl as well as
   all questions and answers in english.",},
{"role": "user",
"content": f'Create as many relevant questions and the
   corresponding answers for this text: {text}. Return each
   question-answer pair in the following format:
   {{"conversations": [<insert question here> , <insert answer
   here>], "source":{degree_level+"_"+degree}}}',},
]
```

Figure 5.2: Prompt used with the ChatGPT API. The first part is a prompt to the role system, instructing it to act as an assistant to help create a dataset. In the second part, the user prompt is inserted, containing the text that is going to be labeled, as well as instructions on how to format the questions and answers.

```
"conversations":  ["Is the Bachelor of Engineering
    {"conversations": ["What identification documents are required
   for exams?", "You are required to bring your student ID card
   for identification. Your ID card and a current certificate of
   enrollment can also be used for identification."],
   "source":"elektrotechnik_Bachelor"}
"""json
"conversations":  ["What are the learning objectives of the
   Praktikum?", "The learning objectives of the Praktikum include
   gaining insight into"`json
```

Figure 5.3: Errors in dataset generation: Sometimes the outputs of a question-answer pair were not complete. Other times random strings like """json were inserted and even interrupted a current output.

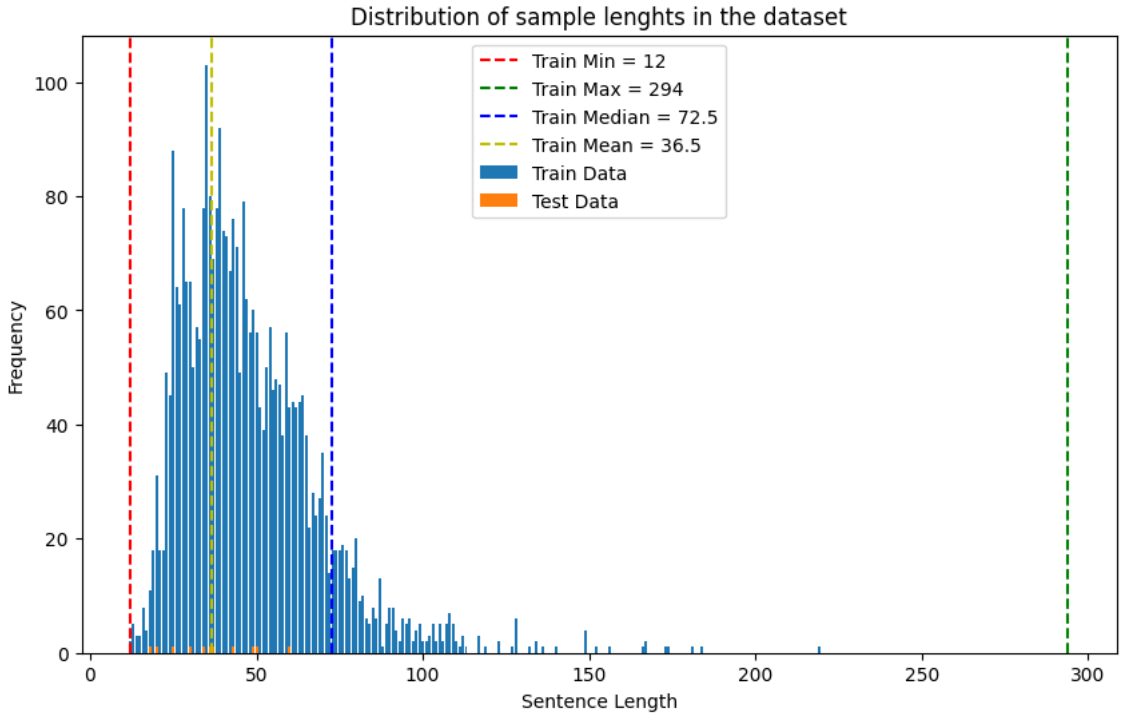

Figure 5.4: Distribution of the sequence lengths in tokens. Each sequence contains a question-answer pair.

6 Implementation

6.1 Used LLM and framework

For this work, the model Mistral-7B-v0.1 from Mistral Ai was used (JSM+23). Figure 6.1 shows the model parameters the authors used. Mistral 7B outperforms larger models like Llama 13B on various benchmarks. To do so multiple new changes were introduced. It uses Sliding Window Attention (SWA) to attend to information beyond the window size of the model. Furthermore, a rolling buffer cache gets used which which significantly reduces the cache memory usage without impacting the model quality, which is useful when processing long sequences. Pre-fill and Chunking get used where the prompt gets pre-filled in the key and value cache and long prompts get chunked to optimize efficiency in generation. The model was fine-tuned using the Lit-GPT framework (AI23) and the already discussed LoRA approach.

Parameter	Value
`dim`	4096
`n_layers`	32
`head_dim`	128
`hidden_dim`	14336
`n_heads`	32
`n_kv_heads`	8
`window_size`	4096
`context_len`	8192
`vocab_size`	32000

Figure 6.1: The Mistral 7B model architecture (JSM+23).

6.2 Model training

The training was done using 6070 iterations on the training dataset. The 6070 iterations reflect the training dataset size of 3035, times two iterations over each sample. In contrast to traditional deep learning where multiple epochs over the same training data are common, iteration too often over the data in fine-tuning results in overfitting and worsens the overall model performance (Ras23). Model training took X minutes. The LoRA rank $r = 8$ with $\alpha = 16$ was used to train the query q, key k and value v projection matrices W_q, W_k, W_v for each transformer layer. LoRA is also able to fine-tune additional layers but this increases the computational requirements, thus only the just mentioned layers were used. The $r = 8$

was used to keep the needed memory usage low. Another useful side effect of using a lower r is that the model does not tend to overfit as with larger r values. This resulted in 4,718,592 trainable parameters, only ,065 % of the total parameter count. The training was done with 32-bit precision, regarded as a default across models and research because it's known to be stable. No quantization was used. The final train dataset contains 3035 question-answer pairs, and the test dataset 159. The high sample count in the training dataset was chosen to give the model as much as possible information, which gets stored in the additional weights. Furthermore, to make the training faster, the test loss was just computed every 100 iterations while training. Thus, the test dataset was sufficiently large to monitor the training process. To minimize memory usage a batch size as well as a micro batch size of one was used. Meaning one sample was processed in one training iteration. The micro-batch size is used to manage memory constraints by subdividing each batch into smaller chunks. AdamW was used as an optimizer which is an extension of the Adam optimizer. The key addition in AdamW is a weight decay mechanism, which helps mitigate overfitting and improves generalization performance. Figure 6.2 shows the loss during training. It can be seen that the loss decreases quickly in the first 100 iterations and then continues to fall only very slightly. Furthermore, it can be seen that the loss seems to be very unstable, this is due to the micro-batch size of 1. The models updates the parameters after every iteration which leads to more unstable loss values. Table 6.1 shows all used hyperparameters.

Figure 6.2: Plot of the model training with 6070 iterations and micro-batch size 1.

Parameter	Value
LoRA rank	8
LoRA alpha	16
Learning Rate	3e-4
Iterations	6070
LoRA dropout	0.05
Weight decay	0.05
Optimizer	AdamW
Batch Size	1
Mini-Batch Size	1

Table 6.1: Hyperparameters used for training

6.3 Computational demands

Model training required approximately 32GB of memory and was performed on a single NVIDIA V100 GPU. Model inference requires only around 14.7 GB of memory, which makes it possible to run on consumer hardware. The final model was trained for 42 minutes with 6070 iterations.

7 Evaluation

For evaluation, BLEU, ROUGE, and perplexity were used as metrics. BLEU measures the similarity between a candidate translation and one or more reference translations. BLEU evaluates the adequacy and fluency of translations, by assigning a score between 0 and 1, where higher scores indicate better translations (PRWZ02). ROUGE measures the similarity between the generated output and one or more references based on the overlap of n-grams, word sequences, and word subsequences (Lin04, LH03). Perplexity measures the effectiveness of a language model in predicting a given sequence of words. A lower perplexity indicates that the language model performs better at predicting the text (JMBB77). Table **??** shows the mean scores computed from the evaluation of the 3 datasets. The training and test datasets have an average BLEU score of around 0.6 which indicates moderate quality of the generation with some overlap but also notable differences from the reference translations. The ROUGE scores for the training and test datasets of around 0.70 suggest that the candidate summary achieves a good level of agreement with the reference summaries. The average perplexity of the training and test dataset is relatively low, but it has to be noted that the perplexity scores also depend on factors such as the size and complexity of the language model, and the nature of the training data. When looking at the evaluation scores the BLEU is almost at zero. This can be explained by the detailed explanations given in the evaluation dataset and the short outputs given by the model (see Figure 7.2). The same goes for the ROGUE score, which is just slightly higher. It has to be mentioned that the BLEU and ROUGE scores alone can't determine the overall quality of the model outputs. The perplexity score is comparable with the training and test dataset, which is only slightly higher than the last two mentioned datasets. Figure 7.1 shows excerpts from the test dataset evaluation results. The input column shows the question inserted into the model, the output column shows the model output and the reference is the actual labeled answer. After that follow the perplexity, BLEU, and ROUGE of the given input. Rows 1, 8, and 11 demonstrate that the model generated the exact answer that was given in the reference, resulting in a BLEU and ROUGE score of 1. The model's high performance can be attributed to the fact that the questions were in the training and the test dataset. Because the data scraped from the study programs contains redundant information some questions and answers were included more than once in the entire dataset. The extracted input from row 3 is from an alumni interview. The output differs from the reference and provides an incorrect answer. The BLEU and ROUGE scores accurately measure this deviation. In row 10, the BLEU and ROUGE scores are low because the answer is phrased differently than the reference, although the output is correct. In this instance, the perplexity score is a more reliable indicator of performance. Row number 11 has a relatively high perplexity score of 3.08, despite the correct output seen in the other scores. This may indicate that the model lacks confidence in generating names

	BLEU	ROUGE	Perplexity
Train	0.62	0.75	2.21
Test	0.54	0.70	2.15
Eval	0.005	0.13	2.82

Table 7.1: Mean of the evaluation scores over the train, test, and evaluation dataset.

that were only learned during fine-tuning. Additionally, it is worth noting that the input and output sentences contain German terms that were not translated during the labeling process, as seen in rows 2, 8, and 11. In summary, the results are mostly accurate, with some deviations from the reference. However, as long as the model output provides relevant information that aligns with the context of the question, these deviations are not critical. The evaluation dataset results are shown in Figure 7.2. It can be concluded that the model produces concise outputs, but lacks details in some cases, as confirmed by the low BLEU and ROUGE scores. Row 1 of the results indicates that the model output does not mention all the study programs included in the scraped data, suggesting potential shortcomings in the proposed dataset. In row 6, the model produces incorrect graduation rate percentages but correctly predicts the employment rate of graduates. Rows 9 and 10 of the evaluation dataset contain questions in German. The model effectively processed German inputs and adapted the output accordingly. The model sometimes gives incorrect or insufficient answers, which for the most part can be improved by expanding and completing the training dataset, because the evaluation has shown that the model is capable of learning the information provided in fine-tuning. The evaluation also showed that using multiple evaluation metrics is important to capture the quality of the model outputs. Due to the generative nature of LLMs, the answers may differ from the provided references without being fundamentally incorrect.

	Input	Output	Reference	Perplexity	BLEU	Rogue
1	What is the purpose of the agreement?	The purpose of the agreement is to regulate the transfer of devices between the recipient and the issuing institute.	The purpose of the agreement is to regulate the transfer of devices between the recipient and the issuing institute.	1.96	1.0	1.0
2	What should be included in the 'Begründung' section of the postponement request?	The 'Begründung' should include a justification for the extension, such as a research stay in a company or abroad, participation in a seminars series, or other commitments that make it impossible to meet the deadline.	The 'Begründung' section of the postponement request should include the reason for the request.	2.14	0.0	0.27
3	What types of activities are included in the Studium Generale at Hochschule Aalen?	The Studium Generale at Hochschule Aalen includes activities such as lectures, one-day seminars, two-day seminars, work in a social institution, voluntary work in committees or defined tasks at Hochschule Aalen, inaugural lectures, and events included in the program of the Studium Generale.	The Studium Generale at Hochschule Aalen includes activities such as lectures, one-day seminars, two-day seminars, work in a social institution, voluntary work in committees or defined tasks at the university, inaugural lectures, and events included in the program of the Studium Generale of the degree programs.	1.88	0.83	0.91
4	How did Barbara Hönle establish contact with her current employer?	Barbara Hönle established contact with her current employer through a recommendation from a former supervisor.	Barbara Hönle currently works in the family business, and her contacts with her previous employer were established through her practical semester and her bachelor's thesis.	2.06	0.0	0.32
5	What aspects need to be coordinated with the supervising professor regarding the registration, submission, scope, and form of the project work?	The registration, submission, scope, and form of the project work need to be coordinated with the supervising professor.	The aspects that need to be coordinated with the supervising professor regarding the registration, submission, scope, and form of the project work are the registration, submission, scope, and form.	1.68	0.44	0.46
6	What occurs if my appeal is granted?	If your appeal is granted, you can continue your studies in your course of study. You will receive a corresponding letter from the Hochschule Aalen. If your appeal is not granted, you will also receive a corresponding letter from the Hochschule Aalen. You will then have the option to file a lawsuit at the Administrative Court in Stuttgart.	If your appeal is granted, you can continue your studies in your program. You will receive a corresponding letter from the Hochschule Aalen. If your appeal is not granted, you will also receive a corresponding letter from the Hochschule Aalen. You will then have the option to file a lawsuit with the Administrative Court in Stuttgart.	1.76	0.88	0.94
7	What are the credit requirements for elective modules in the Informatik program at Hochschule Aalen?	The credit requirements for elective modules in the Informatik program at Hochschule Aalen require a minimum of 5 CP, with at least 3 CP from the options provided by the program, and the remaining 2 CP to be chosen from options provided by the program.	The credit requirements for elective modules in the Informatik program at Hochschule Aalen include specific credit point allocations for different specializations within the program, with a minimum number of credit points required from the options provided by the program and from courses	1.81	0.42	0.57
8	How many Kolloquiums-Terminen is it mandatory to attend before completing the Bachelor thesis module?	It is mandatory to attend 3 Kolloquiums-Terminen before completing the Bachelor thesis module.	It is mandatory to attend 3 Kolloquiums-Terminen before completing the Bachelor thesis module.	2.27	1.0	1.0
9	From when will the functionalities for exams and exam periods be available?	The functionalities for exams and exam periods will be available from the summer semester of 2020.	The mentioned functionalities will also be available for exams and the exam period starting from the summer semester of 2020.	1.99	0.39	0.66
10	Why are machine learning methods becoming increasingly important in applications?	Machine learning methods allow systems to learn from data, improving their performance over time, and this learning ability is crucial for modern applications.	Machine learning methods are becoming increasingly important due to the availability of more data and experiences from various fields, thanks to the global process of digitalization.	1.9	0.0	0.24
11	What are Prof. Dr. Roland Dietrich's Sprechzeiten?	Prof. Dr. Roland Dietrich's Sprechzeiten are nach Absprache.	Prof. Dr. Roland Dietrich's Sprechzeiten are nach Absprache.	3.08	1.0	1.0

Figure 7.1: Excerpt from the test dataset evaluation results, all outputs were generated with temperature = 0.5

	Input	Output	Reference	Perplexity	BLEU	ROUGE
1	What can I study at Aalen University	You can study the following programs at Aalen University: Bachelor of Engineering, Bachelor of Engineering Technology, Bachelor of Business Administration, Bachelor of Arts, Bachelor of Arts & Economics, Bachelor of Design, Bachelor of Media and Information Technology, Master of Engineering, Master of Engineering Technology, Master of Business Administration, Master of Arts, Master of Arts & Economics, Master of Computer Science, Master of Software and Web Technologies, Master of Electrical and Energy Systems, and Master of Mechatronics	"Aalen University, located in Aalen, Germany, offers a wide range of study programs in various fields. Some of the key areas of study at Aalen University include: 1. Engineering: Aalen University offers programs in mechanical engineering, electrical engineering, industrial engineering, and materials science. 2. Business and Management: The university provides courses in business administration, international business, and industrial management. 3. Information Technology: Aalen University offers programs in computer science, software engineering, and information management. 4. Optics and Photonics: The university is known for its expertise in optics and photonics, offering programs in this specialized field. 5. Life Sciences: Aalen University provides courses in biotechnology, medical technology, and pharmaceutical engineering. 6. Energy and Environmental Management: The university offers programs related to sustainable energy and environmental management. These are just a few examples of the diverse range of study programs	1.58	0.0	0.21
2	I would like to give up my studies, how can I do this?	You can give up your studies by submitting a form to the secretariat. The form can be found on the Hochschule Aalen website.	"To give up your studies at Aalen University, you would typically need to follow a formal process. Here are the general steps you may need to take: 1. Contact the Student Services Office: Get in touch with the Student Services Office at Aalen University to inform them of your decision to give up your studies. They will provide you with the necessary information and guidance on the process. 2. Submit a Withdrawal Request: You may be required to submit a formal request for withdrawal from your studies. This request may include personal details, student ID, and the reason for your decision. 3. Fulfill any Obligations: Ensure that you fulfill any outstanding academic or administrative obligations, such as returning library books, settling outstanding fees, or completing any necessary paperwork. 4. Seek Guidance: If you have any concerns or questions about the withdrawal process, consider seeking guidance from academic advisors, counselors, or the Student Services Office. It's important to note that the specific procedures for withdrawing from studies may vary based on your individual circumstances, such as your program, enrollment status, and any applicable regulations. Therefore, it's advisable to directly contact Aalen University's Student Services Office for personalized guidance on the withdrawal process."	1.74	0.0	0.1
3	What are the admission requirements for the university?	The admission requirements include having a high school diploma or equivalent, and a minimum grade of 2.5 in the final grade.	"The admission requirements for Aalen University in Germany vary depending on the program and level of study. However, in general, the following are common admission requirements: 1. Bachelor's Programs: - Secondary school leaving certificate or equivalent - Proof of German language proficiency (if the program is taught in German) - Some programs may have additional entrance exams or specific subject requirements 2. Master's Programs: - A recognized bachelor's degree or equivalent in a related field - Proof of German language proficiency (if the program is taught in German) - Some programs may require relevant work experience or specific prerequisite courses 3. Language Requirements: - For programs taught in German, applicants are typically required to demonstrate proficiency in the German language through exams such as TestDaF or DSH. - For programs taught in English, applicants may need to provide proof of English language proficiency through tests like IELTS or TOEFL. It's important to note that specific admission requirements can vary by program, so it's recommended to check the university's official website or contact the admissions office for the most accurate and up-to-date information."	1.69	0.0	0.12
4	What are the housing options for students?	Students can apply for a room in the university residences or search for private accommodation. The university offers guidance and support in both processes.	"At Aalen University in Aalen, Germany, students have several housing options available to them. The university offers assistance in finding accommodation, including student dormitories, shared apartments, and private rentals. The Studentenwerk Ulm, a student services organization, also provides support in finding suitable housing for students. Additionally, there are private housing options available in the city of Aalen, including apartments and shared living spaces. The university's International Office can provide further information and guidance on housing options for international students."	1.69	0.01	0.15
5	How is the university's reputation?	The university has a good reputation, especially in the field of computer science.	"Aalen University, located in Aalen, Germany, has a strong reputation for its focus on applied sciences and engineering. The university is known for its practical and hands-on approach to education, with a strong emphasis on industry collaboration and real-world application of knowledge. Aalen University is well-regarded for its research activities, particularly in the fields of materials science, optics, and photonics. Additionally, the university has a good reputation for its international partnerships and exchange programs, providing students with opportunities for global learning experiences. Overall, Aalen University is respected for its commitment to providing high-quality education and producing graduates with strong practical skills and industry relevance."	1.7	0.0	0.13
6	What is the graduation rate and what percentage of students find employment after graduation?	The graduation rate is 97%, and 90% of graduates find employment.	"As of the latest available data, the graduation rate at Aalen University in Aalen, Germany is approximately 70%. Additionally, around 90% of Aalen University graduates find employment within six months of completing their studies. These figures demonstrate the university's commitment to providing a high-quality education and preparing students for successful careers."	1.9	0.0	0.28
7	What student organizations or clubs are available on campus?	The campus has various student organizations or clubs, including the Fakultätsrat, Studienberatung, Sportwirtschaftliche Vereinigung, and Studienkolleg Konstanz.	At Aalen University in Aalen, Germany, there are various student organizations and clubs available on campus. These include academic and professional organizations related to different fields of study such as engineering, business, and technology. Additionally, there are cultural and social clubs, sports teams, and interest-based groups that cater to a wide range of hobbies and activities. Some examples of student organizations at Aalen University include the Aalen University Motorsport Team, the Aalen University Business Club, and the Aalen University Music Club. These organizations provide students with opportunities to network, socialize, and engage in extracurricular activities outside of their academic studies.	1.68	0.0	0.1
8	What opportunities are there for study abroad programs?	There are opportunities for study abroad programs, including Erasmus, Hochschule Aalen, and Hochschule Merseburg.	Aalen University in Aalen, Germany offers various opportunities for study abroad programs. The university has partnerships with numerous international universities, allowing students to participate in exchange programs and study abroad experiences. These programs provide students with the chance to study in different countries, immerse themselves in new cultures, and gain valuable international experience. Additionally, Aalen University offers support and guidance for students interested in pursuing study abroad opportunities, including assistance with the application process, visa requirements, and academic planning. Overall, the university encourages and facilitates study abroad experiences to enhance students' academic and personal growth.	1.67	0.0	0.13
9	Wie hoch sind die Studiengebühren für ein Grundstudium/Graduiertenprogramm?	Die Studiengebühr beträgt 186 Euro pro Semester.	Die Studiengebühren an der Hochschule Aalen variieren je nach Studiengang und der individuellen Studiensituation. Generell gibt es in Deutschland keine allgemeinen Studiengebühren für grundständige Studiengänge an staatlichen Hochschulen. Für Graduiertenprogramme können jedoch Studiengebühren anfallen, die je nach Programm unterschiedlich ausfallen können. Es wird empfohlen, sich direkt an die Hochschule Aalen zu wenden, um genaue Informationen zu den Studiengebühren für spezifische Studiengänge und Graduiertenprogramme zu erhalten.	2.05	0.0	0.05
10	Wie kann ich mich in einem Studiengang einschreiben?	Sie können sich an den Leiter des Studiengangs oder die Prüfungsausschussvorsitzende wenden.	Um sich an der Hochschule Aalen für einen Studiengang einzuschreiben, müssen Sie zunächst die Zulassungsvoraussetzungen für den gewünschten Studiengang erfüllen. Anschließend müssen Sie sich online über das Bewerbungsportal der Hochschule Aalen bewerben. Dazu müssen Sie alle erforderlichen Unterlagen einreichen, wie zum Beispiel beglaubigte Zeugniskopien, Lebenslauf, Motivationsschreiben und ggf. Sprachnachweise.	2.12	0.0	0.06

Figure 7.2: Excerpt from the evaluation dataset results, all outputs were generated with temperature = 0.5

8 Summary and Discussion

This paper describes the development of a custom dataset using publicly available web data. A 7-billion parameter open-source LLM was fine-tuned to create a question-answering chatbot for the Computer Science Department at Aalen University. The paper discusses the history, technical basics, and limitations of LLMs. The data generation process is explained, and the results are evaluated. The paper presents several new findings. The dataset creation process revealed that LLMs, such as ChatGPT, can generate datasets inexpensively. An API, such as the one offered by OpenAI, can automate the dataset creation process without expensive human labeling. However, manual correction was still necessary to rectify some errors. The proposed approach is useful and can produce good results, but requires further improvement in extracting more detailed and relevant answers. The model exhibited strong performance on prompts included in the training set and demonstrated good translation capabilities even for languages it was not specifically fine-tuned on. However, it occasionally struggles to identify the correct answer when presented with slightly modified prompts. Additionally, the answers are mostly brief and tend to lack detail. The most limiting factor is the proposed dataset. The information provided about the computer science department at Aalen University is incomplete and requires expansion. Additionally, the dataset may become outdated without regular updates. To address this, future improvements could include implementing a pipeline for regular data updates and model retraining or integrating an approach like RAG. Furthermore, it is recommended that future work concentrates on creating datasets through open-source LLMs. To guarantee that the model can be executed on computers with limited memory, further testing should be conducted by reducing the model size and experimenting with the parameters.

Bibliography

(AAA+23) Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models, 2023.

(AI23) Lightning AI. Lit-gpt. `https://github.com/Lightning-AI/lit-gpt`, 2023.

(Ala20) Nefi Alarcon. Openai presents gpt-3, a 175 billion parameters language model. Online, 2020. Accessed 12-February-2024.

(ANS23) Deepak Suresh Asudani, Naresh Kumar Nagwani, and Pradeep Singh. Impact of word embedding models on text analytics in deep learning environment: a review. *Artificial Intelligence Review*, 56:10345–10425, 2023.

(A.R23) Mystic AI A.Ram. Diversity in ai: From closed-source ml to open-source innovation. Online, November 2023. Accessed 22-February-2024.

(BMR+20) Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

(Cha24) Natalee Champlin. Unlocking the true potential of openai's custom gpts. Online, February 2024. Accessed 22-February-2024.

(DCLT19) Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. volume 1, page 4171 – 4186, 2019. Cited by: 29499.

(DPHZ23) Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.

(EC21) Zoubin Ghahramani Eli Collins. Lamda: our breakthrough conversation technology. Online, May 2021. Accessed 15-February-2024.

(Fac) Hugging Face. Hugging face. Online. Accessed 23-February-2024.

(Fac23) Hugging Face. 2023, year of open llms. Online, December 2023. Accessed 14-February-2024.

(GD23) Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

(GGS+20) Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *CoRR*, abs/2009.11462, 2020.

(Goo23) Google. Introducing gemini: our largest and most capable ai model. Online, December 2023. Accessed 12-February-2024.

(Goo24) Google. Our next-generation model: Gemini 1.5. Online, February 2024. Accessed 22-February-2024.

(GSS+24) Jonas Geiping, Alex Stein, Manli Shu, Khalid Saifullah, Yuxin Wen, and Tom Goldstein. Coercing llms to do and reveal (almost) anything, 2024.

(GZ21) Eghbal Ghazizadeh and Pengxiang Zhu. A systematic literature review of natural language processing: Current state, challenges and risks. In Kohei Arai, Supriya Kapoor, and Rahul Bhatia, editors, *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1*, pages 634–647, Cham, 2021. Springer International Publishing.

(Hay23) Haystack. Tutorial: Question generation. Online, June 2023. Accessed 25-February-2024.

(HSW+21) Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

(IM23) Régis Pierrard Ilyas Moutawwakil. Llm-perf leaderboard. `https://huggingface.co/spaces/optimum/llm-perf-leaderboard`, 2023.

(JMBB77) Frederick Jelinek, Robert L. Mercer, Lalit R. Bahl, and Janet M. Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *Journal of the Acoustical Society of America*, 62, 1977.

(Jon17) M. Tim Jones. Speaking out loud. Online, 2017. Accessed 15-February-2024.

(JSM+23) Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

(JSR+24) Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szy-

mon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

(KDBY23) Kuldeep Singh Kaswan, Jagjit Singh Dhatterwal, Reenu Batra, and Dileep Kumar Yadav. Chatgpt: A comprehensive review of a large language model. In *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI)*, pages 738–743, 2023.

(KHM+23) Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models, 2023.

(KMH+20) Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

(Lan) LangChain24. Applications that can reason. powered by langchain. Online. Accessed 28-February-2024.

(LFdS+17) Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding, 2017.

(LH03) Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, page 71–78, USA, 2003. Association for Computational Linguistics.

(Lin04) Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

(LPP+21) Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

(LWY+24) Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation, 2024.

(MBP+23) Rajiv Movva, Sidhika Balachandar, Kenny Peng, Gabriel Agostini, Nikhil Garg, and Emma Pierson. Topics, authors, and networks in large language model research: Trends from a survey of 17k arxiv papers, 2023.

(MCCD13) Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

(MES19) Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. Word-class embeddings for multiclass text classification, 2019.

(Mic23a) Microsoft. Microsoft and openai extend partnership. Online, January

2023. Accessed 12-February-2024.

(Mic23b) Microsoft. Phi-2: The surprising power of small language models. Online, December 2023. Accessed 12-February-2024.

(NFA+21) Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization, 2021.

(Oll) Ollama. Get up and running with large language models, locally. Online. Accessed 23-February-2024.

(Ope23) OpenAI. Introducing gpts. Online, November 2023. Accessed 22-February-2024.

(Per23) Billy Perrigo. Meta's ai chief yann lecun on agi, open-source, and ai risk. Online, February 2023. Accessed 15-February-2024.

(PNI+18) Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

(PRWZ02) Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

(Ras23) Sebastian Raschka. Practical tips for finetuning llms using lora (low-rank adaptation). Online, November 2023. Accessed 28-February-2024.

(Rei) Kenneth Reitz. Requests: Http for humans. Online. Accessed 24-February-2024.

(RN18) Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

(RNSS18) Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

(Sou) Beautiful Soup. Beautiful soup documentation. Online. Accessed 24-February-2024.

(Spi23) Arthur Spirling. Why open-source generative ai models are an ethical way forward for science. *Nature*, 616:413, 2023.

(tea23) Mistral AI team. Mixtral of experts, a high quality sparse mixture-of-experts. Online, December 2023. Accessed 15-February-2024.

(TGZ+23) Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

(TLI+23) Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

(TMS+23) Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

(VSP+23) Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

(ZHL+23) Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention, 2023.

(ZLS+24) Travis Zack, Eric Lehman, Mirac Suzgun, Jorge A Rodriguez, Leo Anthony Celi, Judy Gichoya, Dan Jurafsky, Peter Szolovits, David W Bates, Raja-Elie E Abdulnour, Atul J Butte, and Emily Alsentzer. Assessing the potential of gpt-4 to perpetuate racial and gender biases in health care: a model evaluation study. *The Lancet Digital Health*, 6(1):e12–e22, 2024.

(ZLX+23) Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023.

(Zor23) Julia Zorthian. Openai ceo sam altman asks congress to regulate ai. Online, May 2023. Accessed 15-February-2024.