EPiC Series in Computing

Volume 105, 2025, Pages 108–116

Proceedings of EUNIS 2024 annual congress in Athens

Building UniGPT: A Customizable On-Premise LLM-Solution for Universities

Jonathan Radas[1,2], Benjamin Risse[2], and Raimund Vogl[1]

[1] Center for Information Technology (CIT), University of Münster, Germany
[2] Institute for Geoinformatics & Faculty of Mathematics and Computer Science, University of Münster, Germany
{jonathan.radas,b.risse,rvogl}@uni-muenster.de

Abstract

Large Language Models (LLMs) have become the hot topic in Artificial Intelligence (AI) in the last few years, especially with the advent of the Generative Pretrained Transformer (GPT) models and the release of ChatGPT in November 2022. Demand from faculty members for access to such models quickly arose and proved to be hard to address in an orderly manner by central IT providers due to technical, privacy and payment constraints by the major suppliers. Additionally, specific research questions might require more control over the model and deployment. Thus, access to non-public on-premises models became desirable and became possible with open-source solutions (e.g. Llama 2 by Meta and Mixtral 8x7B by Mistral AI). Due to the large model sizes, on-premises deployments are demanding in terms of hardware and system engineering. In this paper, we present our deployment of a large language model for the University of Münster, a service we call UniGPT. We focus on the high-level architecture, consisting of the frontend, backend, and models, and also discuss the experiences with our deployed service.

1 Why did we deploy our own LLM?

Since the release of GPT-3 [1] and the subsequent release of ChatGPT, large language models have gained mainstream adoption. ChatGPT reached over 100 million users just two months after it was made public [2]. Despite its name, OpenAI does not publicly release their most recent large language models (LLMs); instead, they provide access solely through their proprietary ChatGPT interface and API, offered as a software-as-a-service (SaaS) solution. This approach presents significant limitations for our university and similar institutions. In contrast, hosting LLMs locally offers numerous benefits:

More control over the models. This means choosing models specifically for our use cases. With the possibility of hosting arbitrary models, which were not restricted by private companies, potentially even biased and harmful ones, we can enable researchers to conduct their own research with specific models or even do user studies, without requiring them to have a deep understanding of deploying LLMs.

Less dependency on OpenAI. OpenAI controls large parts of the market, with 14.6 billion visits, while Google Bard (now Gemini), its biggest competitor, had just 3.8 billion

R. Vogl, L. Desnos, J.-F. Desnos, S. Bolis, L. Merakos, G. Ferrell, E. Tsili and M. Roumeliotis (eds.), EUNIS 2024 (EPiC Series in Computing, vol. 105), pp. 108–116

visits [3]. OpenAI might change their terms of use, pricing, or their available models. Digital sovereignty can only be accomplished by having control over the AI models ourselves.

LLMs as a shared resource in bigger research consortia. Especially in bigger research consortia such as collaborative research centres, having access to shared digital resources is of utmost importance. LLMs can provide a variety of centre-specific services to accelerate the often interdisciplinary communication, particularly with sensitive data.

Privacy and copyright concerns with OpenAI. The utilization of OpenAI's services raises concerns regarding data protection and privacy compliance, particularly with the General Data Protection Regulation (GDPR), especially when personal data is involved (e.g. student assignments, etc.). Furthermore, transferring information to cloud services without adequate regulations could lead to copyright issues, including potential infringement. Another issue lies in the necessity of creating individual user accounts to utilize ChatGPT or enforce OpenAI's usage terms for enterprise API access, thus enabling direct interaction between members of our university and OpenAI's servers. Additionally, sensitive research data should not be sent to LLM cloud services where prompts are used for training and other downstream uses. Finally, university users might want to process and analyse more sensitive data, e.g. medical data, which is not allowed with off-premises AI cloud services.

1.1 Audience for the UniGPT service

Soon after the introduction of ChatGPT, there was demand from both faculty and students that the university should make free access to this or similar services available. With the possibility to host powerful open source LLMs on-premises, we plan to make this service available for researchers, students, and also for university administration. The interest comes from a wide range of disciplines, even beyond computer science. Interestingly, the first interested parties were researchers from psychology, who want to use this in classes to work with students. Since several universities in Germany are already making (or planning to make) ChatGPT available as a general service for all of their members, it is clear that the scope of our on-premises service must include all user groups. For the time being, we are focusing the UniGPT service on inference only, i.e. we use readily available LLMs that we deploy on our on-premises infrastructure and do not aim at training models ourselves - a task that would (at least currently) be too demanding for the compute resources of the university.

2 The Architecture

From a high-level perspective, the architecture has three main parts (see figure 1): Firstly, **the frontend-server**, which offers a chat-like UI and is used to communicate with the backend. It also manages access, saves previous conversations (using MongoDB), and can potentially access additional information using Retrieval-Augmented Generation (RAG). Secondly, **the large language model (LLM)** itself is responsible for generating text given the user-specified prompt. Thirdly, **the backend-server**, which manages the LLM. It has an API endpoint and forwards the prompts into the model. It also handles loading the model and distributing it to potentially multiple GPUs. Depending on the implementation, the boundaries between the components are rather fluid. The frontend and the backend server might be deployed within one application.

In order to operationalize our application, we used our pre-existing Kubernetes cluster, which relies upon OpenStack and Ceph as its foundation, thereby ensuring that all components

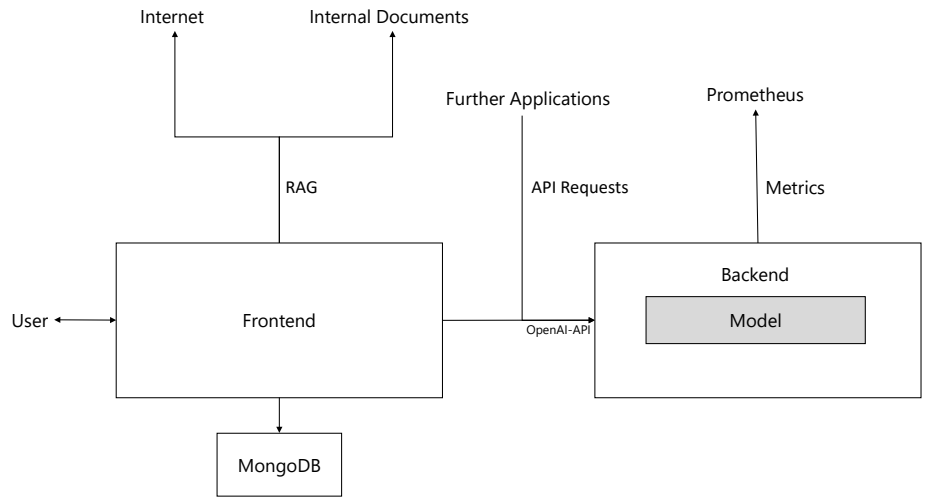

Figure 1: The high-level architecture of our UniGPT service

are built upon robust and reputable open-source software. As hardware to facilitate the necessary computational requirements, a scalable number of NVIDIA A40 graphics processing units (GPUs) were utilized, each equipped with 48 GB of fast memory.

2.1 The frontend

The frontend of a conversationally interactive system serves two primary functions: presenting a user interface and managing the conversations, sessions and users. In the past year, several frontends emerged specifically for conversational AI. Some of the most widely used options include:

- ChatUI by Hugging Face[1]
- text-generation-webui by oobabooga[2].
- LobeChat by LobeHub[3]
- LM Studio by Element Labs[4]

Each of these is adapted to specific use cases. In our initial deployment, we focused on easy accessibility. Most User Interfaces (UI) are aimed at experienced users who are familiar with certain LLM inference settings such as the sampling temperature. Since we assume that most of our students are not familiar with deep learning techniques, we decided to choose a simpler deployment without complicated settings for the user. ChatUI by Hugging Face offers such a solution with a very simple UI but limited options. It follows the basic layout of a chat app

[1] https://github.com/huggingface/chat-ui
[2] https://github.com/oobabooga/text-generation-webui
[3] https://github.com/lobehub/lobe-chat
[4] https://lmstudio.ai/

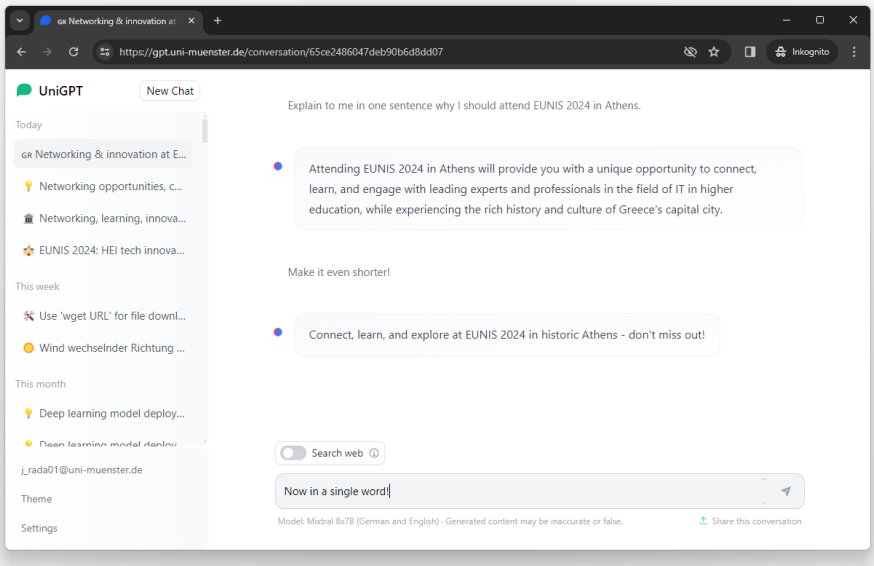

Figure 2: Our frontend with the ChatUI

with previous messages at the top and an input field at the bottom. A screenshot can be seen in figure 2. ChatUI is able to use the OpenAI API for communication with LLM backends - so it is also possible to include external commercial LLM services (like GPT-4, the base for ChatGPT) in the same frontend as alternatives in case of user demands. This component also manages the users and offers authentication and authorization, so it is only available to specific logged-in users from the university (authorization can be managed through user groups). For the authentication, we used the preexisting SATOSA proxy which offers an OIDC interface to Shibboleth. Based on the eduGAIN information about the institution and attributes the access is managed. During the first access to the UniGPT service, the frontend also takes care of having the prospective users to accept the usage policies of Münster University (banning the use of the university-provided AI services to generate and disseminate illegal content and requiring users not to transfer information classified as sensitive to off-premises AI services) and the respective external cloud services that might also be made available in addition to the on-premises LLMs. The frontend also stores the session (using a MongoDB), so old conversations are still available. Clearly, the storage of prior conversations is convenient but also poses a data privacy issue, so users can delete this archive through the settings page of the ChatUI interface.

2.2 The backend

The backends offer multiple functionalities. Directly incorporating PyTorch models into the system is infeasible because it would require allocating memory space for each user, even during periods of prolonged idleness. Instead, multiple users share potentially multiple model shards on multiple GPUs. Moreover, incoming requests from users are managed through a queue mechanism, enabling efficient distribution of computational load. By design, the system operates in a stateless manner; at the time of processing each request (i.e., receiving a new mes-

sage from the user), the preceding interaction history is supplied as input, thereby eliminating the need to maintain state information within the model itself. Additionally, the backend also implements Prometheus metrics and handles dynamic model loading.

There are several backends that offer the OpenAI-API - the de-facto standard - or offer a dedicated wrapper to it. Often used examples are TGI by Hugging Face[5] and llama.cpp by Georgi Gerganov[6]. We decided to use the TGI model, due to simpler communication with the frontend and more standardized PyTorch models under the hood.

2.3 The pretrained large language model

The field of large language models is evolving fast with many open-source models released within the last year, including:

- Mixtral/Mistral [4]: The French startup mistral.ai offers several models: Their smallest model is mistral 7B with 7 billion parameters. With a larger capacity (more parameters), the model can store more information and represent more complex patterns. They also offer a sparse mixture of experts (SMoE) model called Mixtral 8x7B [5] under the Apache 2.0 license. Their larger models are closed-source and only available through their API.
- Llama 2 [6]: Llama 2 is the successor to the Llama 1 model which was also published by Meta AI. It is offered in several sizes: 7B, 13B, 34B and 70B. It is offered under a specific Llama license which allows commercial use for up to 700 million users.
- Falcon [7]: The Falcon models were developed by Technology Innovation Institute, Abu Dhabi. Besides a 7B, and a 40B model, there is also a 180B model available, which is according to its authors the largest openly available model.

As a university with teaching demands often restricted to German, we had the requirement that the model should perform well in German and English. Therefore, we decided to use Mixtral, which was trained on five European languages: English, French, Italian, German, and Spanish. The other models also have some capabilities in German because the training data contain texts in other languages than English. However the proportion was rather low, e.g. the training data for Llama 2 contained only 0.17 % German text, and even lower percentages for other other non-English languages [6].

2.4 Quantization for further performance improvements

In a production environment, the number of tokens generated per second is crucial. Since the available computing power is limited, we want to maximize the throughput, so we can serve as many concurrent users as possible.

Quantization is an important technique in deep learning to improve throughput and memory consumption at the cost of a slight decrease in quality. Quantization means that the precision of the weights (neural network parameters of the model) is reduced. Most models are trained with half-precision floats (FP16) or brain floating point (bfloat16), which are already adapted floats (consuming 16 bits of memory) for neural networks. With quantization techniques, the length is further reduced by converting them to integers with a size of 8 bits or less. Popular algorithms for conversions are AWQ [8] and the preceding GPTQ [9]. We decided to use a 4-bit integer quantization with AWQ, which reduces the model size fourfold, while only marginally reducing

[5] https://github.com/huggingface/text-generation-inference
[6] https://github.com/ggerganov/llama.cpp

quality [8]. This allowed us to deploy 70B parameter models fitting in the GPU memory of the NVidia A40, which offers 48 GB of GPU memory.

2.5 First experiences with UniGPT

To gain insights into the use cases and demands of our users and the associated requirements for our LLM systems, we opened a limited non-advertised test phase of UniGPT in early January of 2024 exclusively for university employees. In the context of a university event on generative AI in teaching and learning planned for late April 2024, this service will be more widely announced and access to students will be enabled. Before deploying UniGPT to the entire university (staff and students), early evaluations are mandatory to identify the most suitable model concerning different dimensions, which presents a nontrivial challenge. The main dimensions are quality and toxicity (see also [6]).

Assessing Model Quality. Although numerous benchmarks exist to evaluate the quality of LLMs (e.g. [10, 11, 12]), certain limitations remain. First of all, some authors tailor the fine-tuning of their models to popular benchmarks. This leads to good results in the benchmark, but potentially subpar performance in real-world applications. Furthermore, most existing benchmarks are solely offered in English, making it challenging to evaluate models for other non-English languages such as German. As our application heavily depends on German language capabilities, finding suitable evaluation methods presents an additional challenge.

Toxicity and Safety Considerations. Evaluating the safety aspect of LLMs adds complexity due to its inherent subjectivity. Determining the appropriate response can vary depending on the context and research objective. For some applications, it might even be useful if the model does not deny toxic or unsafe requests.

3 Outlook: Further use cases within higher education

Beyond the requested use cases from teaching and research in various disciplines, there are other ideas for the use of UniGPT. One capability of LLMs is the ability to summarize text which can help navigate huge amounts of information typically found in a university setting. Especially the integration of a LLM into our literature search engine seems very promising. These approaches of incorporating additional data are known in the literature as Retrieval Augmented Generation (RAG) [13]. Additionally, the full control over the models allows us to better steer the usage, especially for student submissions. It enables us to incorporate watermarks into the generated content and detect them at a later stage should such verification become necessary[14]. Finally, the on-premises provision of LLMs also enables the development of completely new usage scenarios and interdisciplinary research, especially in privacy sensitive domains, e.g. in medicine.

As part of our ongoing efforts, we aim to scale the deployment, to serve the entire university rather than serving selected researchers. Ultimately, we will conduct evaluations of both model and tool selection to ensure consistency with institutional priorities and end-users' needs.

4 Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) – CRC 1450 – 431460824.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[2] Krystal Hu. Chatgpt sets record for fastest-growing user base - analyst note. www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/, 2023.

[3] Sujan Sarkar. Ai industry analysis: 50 most visited ai tools and their 24b+ traffic behavior. writerbuddy.ai/blog/ai-industry-analysis, 2023.

[4] 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. arXiv:2310.06825 [cs.CL], 2023.

[5] 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, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. arXiv:2401.04088 [cs.LG], 2024.

[6] 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. arXiv:2307.09288 [cs.CL], 2023.

[7] 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. arXiv:2311.16867 [cs.CL], 2023.

[8] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. arXiv:2306.00978 [cs.CL], 2023.

[9] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. arXiv:2210.17323 [cs.CL], 2023.

[10] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020.

[11] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a

machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

[12] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[13] 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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

[14] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR, 23–29 Jul 2023.

5 Biographies

J. Radas is a research assistant at the IT department of the University of Münster (CIT). After studying at the University of Melbourne and the University of Münster, he graduated in 2021 with a master's degree in information systems. Currently, he is pursuing a Ph.D. under the supervision of Professor Benjamin Risse, focusing his research on advancing deep learning infrastructure.

B. Risse is a full professor at the University of Münster and the head of the Computer Vision & Machine Learning Systems group. He studied computer science followed by a PhD in the intersection of computer vision, machine learning, and neuroscience. From 2015-2017 he was a postdoctoral researcher at the University of Edinburgh. Apart from his focus on the theory of artificial intelligence (in particular deep learning) he and his group is also interested in interdisciplinary applications of these algorithms.

R. Vogl holds a Ph.D. in elementary particle physics from the University of Innsbruck (Austria). After completing his Ph.D. studies in 1995, he joined Innsbruck University Hospital as IT manager for medical image data solutions and moved on to be deputy head of IT. He served as a lecturer in medical informatics at UMIT (Hall, Austria) and as managing director for a medical image data management software company (icoserve, Innsbruck) and for a center of excellence in medical informatics (HITT, Innsbruck). Since 2007 he has been director of the IT department of the University of Münster (CIT, Germany). His research interests focus on the management of complex information systems and information infrastructures.