Chair of Information Systems and Business Process Management (i17)
Department of Computer Science
TUM School of Computation, Information and Technology
Technical University of Munich

Proposal for Master's Thesis in Information Systems

Furkan Gürbüz

Fine-Tuning Large Language Model with Custom Dataset for Ansible Code Generation

Feinabstimmung eines großen Sprachmodells mit benutzerdefiniertem Datensatz zur Code-Generierung in Ansible

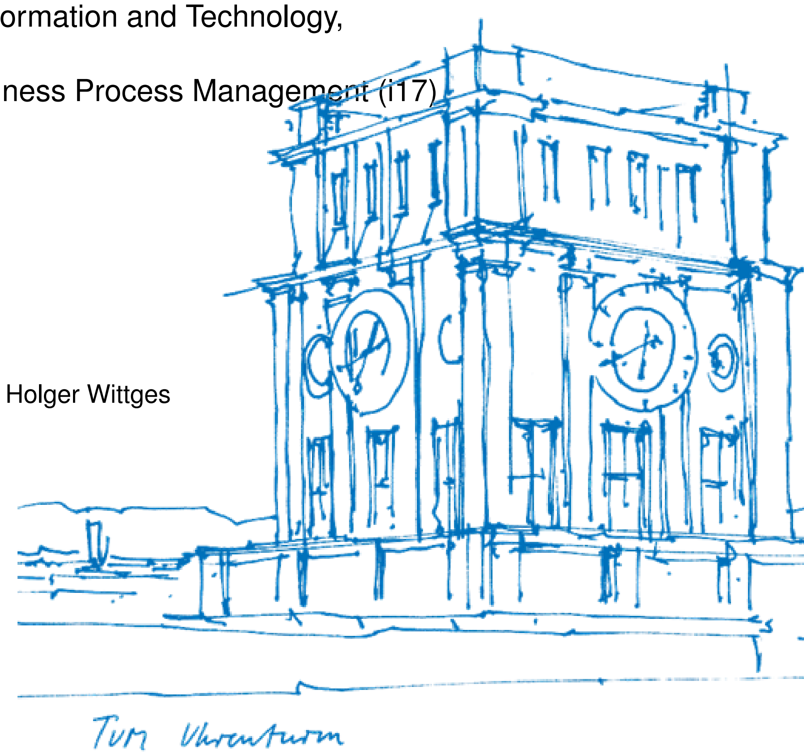
Thesis for the Attainment of the Degree
Master of Science

at the TUM School of Computation, Information and Technology,
Department of Computer Science,
Chair of Information Systems and Business Process Management (i17)

Examiner
Prof. Dr. Stefanie Rinderle-Ma

Supervised by
Thomas Teubner, Maximilian Niedermeier, Dr. Holger Wittges

Submitted on
20.11.2024



Motivation

In the era of digital transformation, the ability to automate IT operations has become a fundamental driver of organizational efficiency and scalability. Modern enterprises and research institutions are increasingly adopting rapid development cycles that demand frequent updates and enhancements (Gupta et al., 2019). This continuous evolution necessitates efficient system management, making the automation of deployments and configurations a critical aspect of DevOps practices (Bass et al., 2015).

Furthermore, Enterprise Resource Planning (ERP) systems are crucial for achieving efficiency and scalability by integrating and streamlining essential business functions (Poston and Grabski, 2000). Among these, SAP stands out as a leading ERP solution, continuously evolving its capabilities since its inception in Germany (Klaus et al., 2000). Educational institutions, such as the SAP University Competence Center at the Technical University of Munich, play a crucial role in providing training and equipping future professionals with the knowledge and skills required to operate ERP systems proficiently (UCC, 2024).

In addition to educating future professionals, the SAP UCC also offer hosting of SAP solutions with robust backup and recovery services. The center utilizes an IT Automation software for installing and configuring SAP systems (UCC, 2024). The software plays a critical role in automating administrative tasks, such as configuration management and application deployment, through the implementation of playbooks. (Ansible, 2024). Manually developing these playbooks is inherently complex and time-consuming, requiring a nuanced understanding of the implementation syntax and the specific requirements of SAP systems (Geerling, 2015). This complexity poses a substantial challenge for organizations aiming to scale operations or fully embrace automation swiftly.

Consequently, there is a substantial demand for code generation and support in implementing Ansible playbooks. As it stands, the academic chair does not possess adequate tools for code generation support in Ansible, and existing models do not meet the requirements satisfactorily. By elevating automation capabilities, IT professionals will be well-equipped to optimize operations. This initiative aligns with trends with focus on enhancing operational efficiency, reducing costs, and accelerating deployment timelines.

Purpose of the Thesis

The goal of this thesis is to optimize the implementation of Ansible playbooks through the use of a fine-tuned large language model. By generating Ansible code specifically tailored for SAP systems, this thesis aims to reduce the time and expertise required for playbook development, making the process more efficient and accessible for both developers and system administrators. This will enable them to focus on higher-level tasks and strategic decision-making. This approach will lead to more consistent, error-free configurations and contribute to the streamlined management of complex IT systems.

In order to achieve this objective, the thesis will involve fine-tuning a large language model specifically for Ansible code generation. The process will begin with the creation of a custom dataset containing Ansible code specifically designed to meet the unique requirements of SAP system configurations. Once the dataset is created, it will undergo preparation and preprocessing to ensure consistency and remove noise. Then, the dataset will be used to train the model, enabling it to generate high-quality, context-specific Ansible playbooks (Howard and Ruder, 2018). By using this custom dataset, the model will learn to produce accurate and efficient Ansible YAML code.

Afterwards, the fine-tuning process follows several key steps, beginning with the selection of a pre-trained model. The model's parameters will be fine-tuned by training it on the custom dataset, adjusting the model to generate code that is syntactically correct and functionally effective. The fine-tuning iterations will involve gradually adjusting hyperparameters and retraining the model to optimize its performance. Upon completion of the fine-tuning process, the model's performance will be tested using a separate validation set to assess its accuracy and effectiveness.

The expected outcomes include not only time and cost savings but also the ability to scale automation across more complex IT environments. Through the integration of advanced machine learning techniques, this research will contribute to the ongoing evolution of IT automation practices. The following section will present the research questions addressed in this thesis, which leans on the Design Science Research methodology.

Research Questions

Research Question 1: What methodology can be used to gather and prepare a custom dataset for fine-tuning large language models?

Methodology:

In the context of this research question the focus is on the creation of a custom dataset, which will be used for fine-tuning the LLM. We employ a Question Answering (QA) dataset format, utilizing specific tools to gather our dataset. Following this, we will clean and preprocess the data to eliminate noise and ensure consistency in data quality and format. This step is crucial for optimizing the dataset for model training. The data needs to be partitioned into different types of set. This will ensure robustness and reliability of the trained model for our use case.

Expected Results:

We have created a dataset, that will be cleaned and prepared for fine-tuning the LLM.

Research Question 2: What steps are involved in implementing the fine tuning process for the large language model?

Methodology:

In the context of this research question the focus is on the fine-tuning process with the created dataset from RQ1. This research question will focus on the implementation and execution of the fine tuning strategies.

Expected Results:

We have tuned the large language model, that is suitable for our use case.

Research Question 3: To what extend does the fine-tuned large language model meet the requirements in terms of performance, accuracy, and applicability?

Methodology:

Concerning this research question the focus is on testing the model performance. Consequently, we test our model's performance with professionals by ensuring the usability, accuracy and functionality.

Expected Results:

We have tested our model and give insights on the performance of our fine tuned LLM in the context of code generation.

Bibliography

Ansible. (2024). Ansible documentation [Accessed: June 25, 2024]. https://docs.ansible.com/ansible/latest/index.html

Bass, L., Weber, I., & Zhu, L. (2015). *Devops: A software architect's perspective*. Addison-Wesley Professional.

Geerling, J. (2015). *Ansible for devops: Server and configuration management for humans*. Leanpub.

Gupta, S., Qian, X., Bhushan, B., & Luo, Z. (2019). Role of cloud erp and big data on firm performance: A dynamic capability view theory perspective. *Management Decision*, *57*(8), 1857–1882.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification.

Klaus, H., Rosemann, M., & Gable, G. G. (2000). What is erp? *Information systems frontiers*, *2*, 141–162.

Poston, R., & Grabski, S. (2000). The impact of enterprise resource planning systems on firm performance.

UCC, S. (2024). Sap ucc documentation [Accessed: October 10, 2024]. https://ucc.tum.de/