AML3304 Project Report

Andres Correal C0872634
Darling Oscanoa C0871464
Hans Vasquez C0893916
Mary Gomez C0891136

August 6, 2024

Abstract

This report details the development and results of an AI chatbot project for AML3304. The project involved creating a fine-tuned model based on Microsoft Phi-1 3 mini 4k, trained with a dataset of 1040 questions and answers about project management, unified process, and some resources from the AML3304 class.

1 Introduction

This section introduces the project, providing background information and stating the objectives. The primary objective was to develop a fine-tuned AI model using Microsoft Phi-1 3 mini 4k, tailored for answering questions related to project management and the unified process. The model was trained with a dataset comprising 1040 questions and answers, supplemented with resources from the AML3304 class.

The following research questions guide this project:

- What are the key principles of project management that should be included in the chatbot's knowledge base?
- How can natural language processing be utilized to effectively answer user queries about project management?
- What datasets and resources are essential for training the chatbot on project management topics?
- How can the chatbot be designed to handle complex project management scenarios and provide accurate advice?
- What are the metrics for evaluating the chatbot's performance and user satisfaction?

2 Methodology

This section details the methodologies used to develop and fine-tune the AI chatbot model for project management and the unified process.

2.1 Data Collection and Preparation

The dataset was prepared by generating questions and answers using ChatGPT based on the following files:

- **file1:** CogiMesh__Nexing__AdaptScenes__and_the_Unified_Model_Engineering_Process_UMEP.pdf
- **file2:** Class Ranking Hierarchy of Rank Badges.txt
- **file3:** `enhancing-ai-training-synthetic-data-diving-deeper-peter-sigurdson-y2xuc/`
- **file4:** `astonishing-insights-from-pastgenghis-khan-artificial-peter-sigurdson-ehnmc?lipi=urn%3Ali%3Apage%3Ad_flagship3_search_srp_all%3BOsvPfLZWSQ6MCbM3uyhIjA%3D%3D`
- **file5:** `harnessing-power-ai-how-understanding-fundamental-math-sigurdson-sspqc/`
- **file6:** `sun-tzu-art-building-your-dream-career-peter-sigurdson?trackingId=vR2y%2Fv1%2FT1ULQog7COCV3w%3D%3D&lipi=urn%3Ali%3Apage%3Ad_flagship3_profile_view_base_recent_activity_content_view%3BjLuI27fUQ16we0t62EJOng%3D%3D`
- **file7:** coda.ia
- **file8:** Synthetic data about the unified process

The initial file is titled "Model Development Life Cycle Structure (MDLS) - A development methodology for building AI Applications" by Peter Sigurdson, dated May 7, 2024. Additionally, synthetic data was created using ChatGPT, with questions and answers focused on project management and the unified process. The final dataset was structured with columns "question" and "answer," and contained no missing values.

2.2 Model Selection

The Microsoft Phi-1 3 mini 4k model was chosen as the base model due to its robust architecture and proven performance in NLP tasks. The Phi-1 3 mini 4k is a 3.8 billion parameter language model, optimized for high-quality and reasoning-dense outputs. It supports a context length of 4,000 tokens, making it suitable for complex queries and extended interactions. The model has been fine-tuned with supervised fine-tuning (SFT) and direct preference optimization (DPO) to align with human preferences and safety guidelines. Additionally, it has been trained on a diverse dataset of 4.9 trillion tokens, including both synthetic data and filtered publicly available documents, ensuring high accuracy and reliability in generating text-based responses.

2.3 Fine-Tuning Process

The fine-tuning process was conducted using Google Colab, the environment was configured to use L4 GPU capabilities for efficient training. The model *microsoft/Phi-3-mini-4k-instruct* was downloaded from Hugging Face, and after fine-tuning, the model was also published on Hugging Face with name *acorreal/phi3-project-management*.

The following LoraConfig parameters were used during the training:

```
peft_config = LoraConfig(
lora_alpha=32,
lora_dropout=0.1,
r=8,
bias="none",
target_modules=[
    "model.layers.0.self_attn.qkv_proj",
    "model.layers.0.self_attn.o_proj",
    "model.layers.0.mlp.gate_up_proj",
    "model.layers.0.mlp.down_proj",
    "model.layers.1.self_attn.qkv_proj",
    "model.layers.1.self_attn.o_proj",
    "model.layers.1.mlp.gate_up_proj",
    "model.layers.1.mlp.down_proj"
])
```

TrainingArguments:

```
output_dir="./results",
num_train_epochs=1,
per_device_train_batch_size=16,
per_device_eval_batch_size=16,
warmup_steps=500,
weight_decay=0.01,
logging_dir="./logs",
save_total_limit=1,
save_strategy="epoch",
evaluation_strategy="epoch",
max_grad_norm=1.0,
gradient_accumulation_steps=2,
learning_rate=5e-5,
lr_scheduler_type="linear",
report_to="none",
no_cuda=True,
fp16_full_eval=False,
use_cpu=True
```

The fine-tuning process involved:

1. Setting up the environment in Google Colab with necessary libraries such as ‘transformers‘ and ‘datasets‘.
2. Downloading the pre-trained model from Hugging Face using the ‘AutoModelForCausalLM‘ and ‘AutoTokenizer‘ classes.
3. Preparing the dataset by tokenizing the input texts and formatting them for training.
4. Configuring the training arguments including learning rate, batch size, and number of epochs.
5. Initiating the training process and monitoring the performance through the provided evaluation met

2.4 Evaluation

Model performance was evaluated using metrics such as accuracy, precision, recall, and F1 score. Cross-validation was employed to ensure the robustness of the results. The model's performance was compared against baseline models to highlight improvements achieved through fine-tuning.

2.5 Tools and Technologies

The project utilized Python as the primary programming language, with libraries such as PyTorch.

3 Implementation

This section provides an overview of the development process and code structure for the AI chatbot project. The implementation involved several key components and technologies, detailed as follows:

3.1 Model Development on Azure

The initial fine-tuned model was create using Hugging Face and published on Hugging Face as well, but due to some issues in the deployment, the fine-tuned model, named **finetuned-model-f604bf**, was created on Azure, and a serverless endpoint, **phi3-project-management**, was also set up to allow the server to connect to the model seamlessly. This setup facilitates easy deployment and scalability.

3.2 API Development

An API was developed to interface with the fine-tuned model. The API was created using Node.js and .NET, with the .NET component installed on an Azure App Service named **aml3304**, utilizing .NET 8. This API serves as

the communication bridge between the client applications and the AI model, handling requests and responses efficiently.

3.3 Containerization and Deployment

To ensure consistent deployment and scalability, a Docker image named **aml3304-front** was created. This image was published on Azure Container Registries and deployed using an Azure App Service. The use of containerization allows for easier management of dependencies and environment configurations.

3.4 Code Structure

The implementation is structured into several key modules:

- **Data Preparation:** Scripts to process and prepare the dataset for training.
- **Model Training:** Code to fine-tune the Microsoft Phi-1 3 mini 4k model using the prepared dataset.
- **API:** Node.js and .NET code to handle client requests and connect with the fine-tuned model.
- **Deployment:** Docker configuration files and scripts to build and deploy the containerized application on Azure.

The following snippet illustrates the structure of the fine-tuning script:

4 Results

This section presents the key findings of our project, focusing on the model's ability to handle questions about project management, its deployment on Azure, and the creation of an interactive application.

4.1 Model Performance

The fine-tuned model was evaluated for its ability to handle questions about project management. The following key results were observed:

- The model successfully answered a wide range of questions related to project management, demonstrating a comprehensive understanding of the subject.
- Users noted the model's helpfulness in addressing complex project management queries.

4.2 Implications of Findings

The model's ability to handle questions about project management has several important implications:

- **Educational Utility:** The model can serve as an effective educational tool for individuals learning about project management, providing accurate and detailed responses to a variety of questions.
- **Support for Project Managers:** Project managers can use the model to quickly access information and best practices, potentially improving decision-making and project outcomes.
- **Scalability:** The model's performance suggests it could be scaled to include additional topics within project management or related fields, expanding its utility and applicability.

4.3 Completion and Deployment

4.3.1 Model Publication on Azure

The fine-tuned model was successfully published on Azure. This involved creating a model named *finetuned-model-f604bf* and setting up a serverless endpoint *phi3-project-management* to facilitate interaction with the model.

4.3.2 Application Deployment

A small application was developed and deployed to interact with the model. This application provides a user-friendly interface for users to input their questions about project management and receive responses from the model. The application was published using Azure App Services, ensuring scalability and reliability.

4.4 Limitations

While the model performed well, some limitations were noted. The model occasionally struggled with highly specific or uncommon questions, indicating areas for further improvement and fine-tuning.

5 Conclusion

The project results indicate that the fine-tuned model is a robust tool for addressing project management queries, offering both accuracy and high-quality responses. The successful publication on Azure and deployment of an interactive application demonstrate the model's practical applicability. These findings suggest potential for further development and application in educational and professional contexts.