Intel Unnati Industrial Training 2024

Kalinga Institute of Industrial Technology (KIIT)

Project Report

Submitted by:

Vansh Motiramani

Under Guidance of:

- Dr. Abhishek Ray
- Abhishek Nandy

Problem Statement:

Introduction to Generative AI and Simple LLM Inference on CPU and finetuning of LLM model to create a Custom Chatbot

Description:

This problem statement is designed to introduce beginners to the exciting field of Generative Artificial Intelligence (GenAI) through a series of hands-on exercises Participants will learn the basics of GenAI, perform simple Large Language Model (LLM) inference on a CPU, and explore the process of finetuning an LLM model to create a custom Chatbots.

Mistral Alpaca Finetuned Chatbot:

GitHub Project

Introduction:

Generative AI is artificial intelligence capable of generating text, images and other types of content. What makes it a fantastic technology is that it democratizes AI, anyone can use it with as little as a text prompt, a sentence written in a natural language. The applications and impact for this is huge, you write or understand reports, write applications and much more, all in seconds.

Large Language Models (LLM):

A large language model, such as Mistral-7B-Instruct-v0.1 from MistralAI, is an advanced artificial intelligence designed to understand and generate human-like text based on vast amounts of training data. They excel at understanding and generating human language. This includes tasks like answering questions, summarizing text, translating languages, and even engaging in natural conversations. They use transformer-based architectures, which allow them to process and generate text in a way that captures complex linguistic patterns and relationships.

Mistral-7B-Instruct-v0.1 :

The LLM model used for this project is Mistral-7B-Instruct-v0.1 by Mistral AI because:

- Outperforms Llama 2 13B on all benchmarks
- Outperforms Llama 1 34B on many benchmarks
- Approaches CodeLlama 7B performance on code, while remaining good at English tasks
- Uses Grouped-query attention (GQA) for faster inference

Dataset:

The dataset used to train the AI model consists of a large collection of labeled data, such as images, text, or audio, that is carefully curated and preprocessed to ensure accuracy, representativeness, and lack of biases. The dataset is divided into training, validation, and test sets, with the training set used to train the model, the validation set used to monitor its performance during training, and the test set used to evaluate the final model. The model's performance is evaluated on the validation set, and the model is fine-tuned and optimized based on this feedback.

Stanford Alpaca Dataset:

The dataset used for this project is Stanford Alpaca dataset is a large-scale NLP dataset developed by researchers at Stanford University. It consists of over 52,000 human-written dialogues between humans and an AI assistant. The dataset covers a wide range of topics and is designed to serve as a benchmark for evaluating language model performance in interactive conversations. The dataset features diverse dialogues, high-quality annotations, and a focus on scalability and ethical considerations. The Alpaca name evokes the friendly and approachable nature of the AI assistant. The dataset has been widely used in the NLP research community for developing and evaluating conversational AI applications.

Overview:

This chatbot represents a powerful integration of the Mistral 7B LLM model and the Stanford Alpaca dataset, resulting in a versatile and capable conversational agent that can understand and follow instructions, engage in open-ended dialogue, and assist users with a wide range of tasks.

Features:

1. Skilled at following instructions and completing tasks
2. Broad knowledge base for engaging conversations on various topics
3. Analytical and research capabilities to provide relevant insights
4. Creative collaboration abilities, including idea generation and feedback
5. Adaptable communication style to suit user preferences
6. Multilingual support for interactions in different languages

Process Flow:

1. **Data Collection:** Obtain the necessary data and resources such as the Stanford Alpaca Dataset to train the chatbot, which contains large corpus of high-quality conversational data.
2. **Data Preprocessing:** Tokenization of the data by the training model that could be interpreted by the model.
3. **LLM Model:** The base LLM model to train the data is necessary for the comprehensive language understanding capability. The Mistral AI-7B Instruct model interprets and process the generated tokens through the model to produce the desired output

Process Flow:

4. **Fine Tuning:** Train the model using pre-processed Stanford Alpaca Dataset.

Step	Training Loss
100	1.427500
200	1.260800

```
8]: TrainOutput(global_step=250, training_loss=1.3197170867919923, metrics={'train_runtime': 1377.191
    7, 'train_samples_per_second': 1.452, 'train_steps_per_second': 0.182, 'total_flos': 3117947529461
    76.0, 'train_loss': 1.3197170867919923, 'epoch': 0.4})
```

Process Flow:

5. User Query Input:

```
from peft import AutoPeftModelForCausalLM
from transformers import GenerationConfig
from transformers import AutoTokenizer
import torch
tokenizer = AutoTokenizer.from_pretrained("/content/mistral-finetuned-alpaca")

inputs = tokenizer("""###Human: What can GenAI do and how it could help world become a better place
```

Update the query accordingly

6. **Server-side processing:** The query is sent through Hugging Face transformers library by accessing the API and then processed through the finetuned model to generate the output. The output is received through the server in the tokens which are then converted to the natural language.

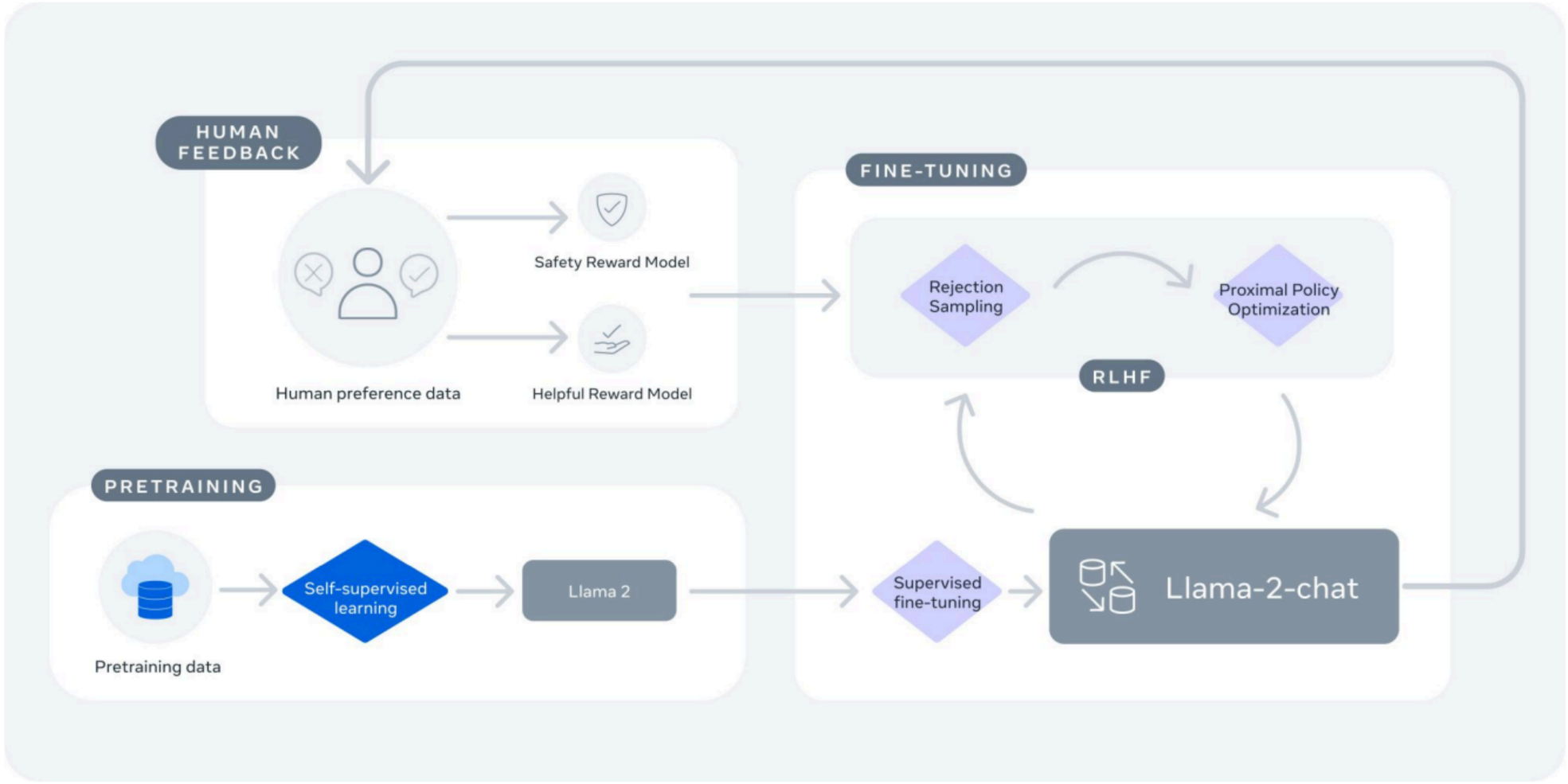
Process Flow:

7. **Displaying the result:** The result is displayed in the natural language:

```
import time
st_time = time.time()
outputs = model.generate(**inputs, generation_config=generation_config)
print(tokenizer.decode(outputs[0], skip_special_tokens=True))
print(time.time()-st_time)
```

```
###Human: What can GenAI do and how it could help world become a better place? ###Assistant: 1. GenA
I can help automate repetitive tasks, freeing up time for more important tasks.
2. GenAI can help improve decision making by providing insights and recommendations.
3. GenAI can help improve customer service by providing personalized recommendations.
4. GenAI can help improve safety by detecting potential hazards and alerting people.
5. GenAI can help improve efficiency by optimizing processes and reducing waste.
6. GenAI can help improve accuracy by
110.82030248641968
```

System Architecture:



Technology used:

- **Environment/Platform:** Intel Developer Cloud Console, Google Colab or Kaggle can be used to run the jupyter notebook by setting up a python environment kernel.
- **Programming Language:** Python language is used throughout the notebook.
- **Core Components :**
  - **Fine tuned Large Language Model**
    - **Mistral 7B Instruct and Alpaca Dataset-** Mistral 7B Instruct large language model fine tuned on Alpaca Dataset

Technology used:

- **Framework:**
  - **Transformers -** hugging face transformers library
    - **AutoModelForCausalLM**: Loads pre-trained causal language models.
    - **AutoTokenizer**: Loads corresponding tokenizers for text preprocessing
    - **BitsAndBytesConfig**: Configures quantization (optional) to reduce memory usage.
    - **HfArgumentParser**: Parses command-line arguments.
    - **TrainingArguments**: Defines parameters for fine-tuning.
    - **pipeline**: Simplifies model inference.
    - **logging**: For logging information and debugging
  - **Pytorch:** PyTorch is an open-source machine learning library based on the Torch library.

Technology used:

- **Framework:**
  - **peft (Parameter-Efficient Fine-Tuning)**: Enables efficient fine-tuning.
  - **LoraConfig**: Configures LoRA for parameter-efficient fine-tuning.
  - **PeftModel**: Wraps the original model for applying PEFT techniques.
  - **trl (Transformer Reinforcement Learning):** Provides tools for reinforcement learning, though used here for supervised fine-tuning (SFT)
  - **SFTTrainer**: Implements the supervised fine-tuning process.
  - **Auto-gptq:** An easy-to-use LLMs quantization package
  - **py7zr:** py7zr is a library and utility to support 7zip archive compression, decompression, encryption and decryption written by Python programming language.

Conclusion:

The chatbot project utilized a range of advanced technologies, including Natural Language Processing Techniques, Hugging Face transformer hubs, Stanford Alpaca Dataset, pytorch libraries.

By integrating these diverse technologies, the project created an intelligent and user-friendly conversational agent with scalable, reliable, and sustainable capabilities.

The Intel Unnati training program provided a valuable foundation for understanding the basics of Generative AI, which helped guide the development of this chatbot project. By applying the principles and techniques learned from the Unnati Program, the team was able to effectively integrate these diverse technologies