INTEL UNNATI INDUSTRIAL TRAINING PROGRAM – 2024

PROJECT REPORT

Submitted By:

LogiCoders

Problem Statement

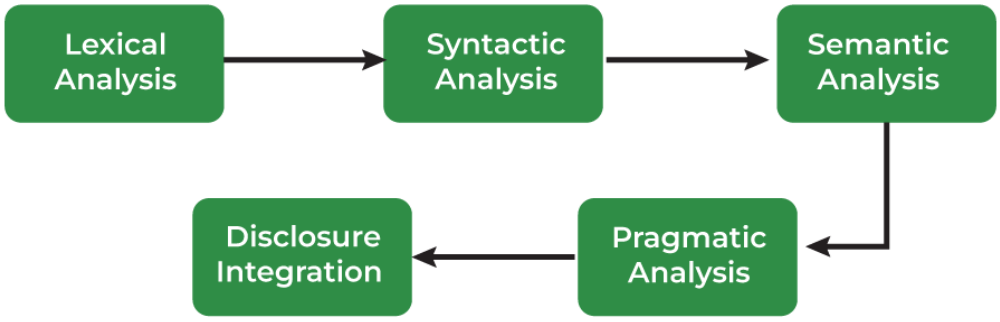
PS-4: “Introduction to GenAI and Simple LLM Inference on CPU and finetuning of LLM Model to create a Custom Chatbot”

Introduction:

This project focuses on creating a custom chatbot using pretrained Large Language Models (LLMs) and perform finetuning with a custom dataset and LLM inference on a CPU. This project explores the fundamental concepts of GenAI, LLM, Finetuning, Inference, Datasets, Transformers, Text-Generation & Chatbots. The project uses Intel Extension for Transformer’s Neural Chat to Finetune a pretrained LLM like “Llama-2-7b-chat-hf” for specific use cases using Alpaca Dataset & Enhance the performance on CPU using Intel Developer Tools, which helps in creating a custom Chatbot for Text-Generation. The Execution & Testing of the Chatbot is done in the Intel Developer Cloud (IDC) Environment on the 4th Generation Intel Xeon Scalable Sapphire Rapid Processor.

Natural Language Processing (NLP):

Natural language processing (NLP) is a subfield of computer science and artificial intelligence (AI) that uses machine learning to enable computers to understand and communicate with human language. NLP enables computers and digital devices to recognize, understand and generate text and speech by combining computational linguistics—the rule-based modeling of human language—together with statistical modeling, machine learning (ML) and deep learning.



Steps in Natural Language Processing

Generative AI (GenAI):

Generative AI refers to deep-learning models that can generate high-quality text, images, and other content based on the data they were trained on. Generative AI refers to deep-learning models that can take raw data and “learn” to generate statistically probable outputs when prompted. At a high level, generative models encode a simplified representation of their training data and draw from it to create a new work that’s similar, but not identical, to the original data.

For example, GPT-3.5, a foundation model trained on large volumes of text, can be adapted for answering questions, text summarization, or sentiment analysis. DALL-E, a multimodal (text-to-image) foundation model, can be adapted to create images, expand images beyond their original size, or create variations of existing paintings.

Large Language Models (LLMs):

A large language model (LLM) is a type of Artificial Intelligence program that can recognize and generate text, among other tasks. LLMs are trained on huge sets of data — hence the name "large."

In simpler terms, an LLM is a computer program that has been fed with enough examples to be able to recognize and interpret human language or other types of complex data. Many LLMs are trained on data that has been gathered from the Internet — thousands or millions of gigabytes' worth of text.

But the quality of the samples impacts how well LLMs will learn natural language, so an LLM's programmers may use a more curated data set.

LLMs use a type of machine learning called deep learning in order to understand how characters, words, and sentences function together. Deep learning involves the probabilistic analysis of unstructured data, which eventually enables the deep learning model to recognize distinctions between pieces of content without human intervention.

LLMs are then further trained via tuning: they are fine-tuned or prompt-tuned to the particular task that the programmer wants them to do, such as interpreting questions and generating responses, or translating text from one language to another.

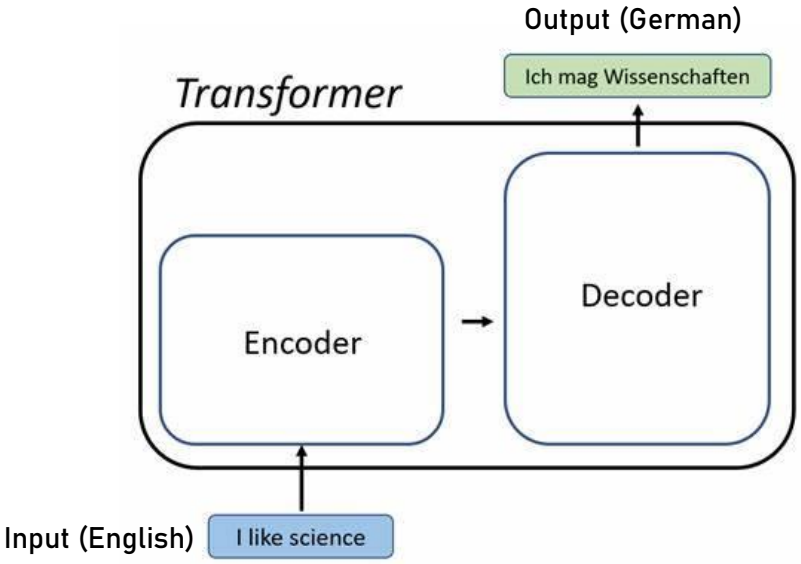
Here is a list of some of the most important areas where LLMs are used:

- **Text generation**
- **Content summarization**
- **AI assistants**
- **Code generation**
- **Sentiment analysis**
- **Language**

Transformers:

LLMs are built on machine learning: specifically, a type of neural network called as the “transformer model”.

A transformer model is a neural network that learns the context of sequential data and generates new data out of it. Transformers are a NLP model and are considered as the evolution of the encoder-decoder architecture. They are specifically designed to comprehend context and meaning by analysing the relationship between different elements, and they rely almost entirely on a mathematical technique called as “attention” to do so.



Simplified Transformers Architecture

Hence transformers are used to handle long range dependencies for understanding the context of conversations. It weighs the importance of different words in a sentence. They also provide real-time responses for handling long conversations and can interact to perform various tasks.

The tokenizer helps in converting the raw text data into individual units, known as “tokens” so that computational models can easily analyse them.

Intel Extension for Transformers:

Intel® Extension for Transformers is an innovative toolkit designed to accelerate GenAI/LLM everywhere with the optimal performance of Transformer-based models on various Intel platforms. It provides model compression experience using Hugging Face Transformers APIs and Intel Neural Compressor. Also provides a Neural Chat Framework to create flexible and powerful chatbots.

Dataset:

A dataset is a structured collection of data that can be used to provide additional context and information to the chatbot. The chatbots have access relevant data and use it to generate responses based on user input. In this project we have used the "Stanford Alpaca" Dataset. This instruction data can be used to conduct instruction tuning for language models and make the language model follow instruction better. The current Alpaca model is fine-tuned from a 7B LLaMA (Large Language Model Meta AI) model on 52K instruction-following data generated by the techniques in the Self-Instruct paper. To reduce the complexity of this large dataset we only use the first 8,000 rows of the alpaca dataset.

Model Used:

The "Llama-2-7b-chat-hf" model has been used for creating the chatbot for text generation in this project. The Llama 2 LLM models are a collection of pretrained & fine-tuned generative text models ranging from 7 billion to 70 billion parameters. The reason for choosing this particular fine-tuned model is that the Llama-2-chat LLMs are designed to understand and generate human-like responses in conversations. They are especially used in chatbots and virtual assistants. It generates a variety of responses and Inferencing is faster. The model can be accessed from HuggingFace using the Intel Extension for Transformers library.

Hardware Configuration:

The Chatbot is fine-tuned and created in the Intel Developer Cloud on the 4th Generation Intel® Xeon® Scalable Processors – Sapphire Rapids.

To get the Hardware configuration of the CPU, we use the "lscpu" command,

```
[redacted]@idc-training-gpu-compute-16:~$ lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         46 bits physical, 57 bits virtual
  Byte Order:            Little Endian
CPU(s):                  192
  On-line CPU(s) list:   0-191
Vendor ID:               GenuineIntel
  Model name:            Intel(R) Xeon(R) Platinum 8468V
    CPU family:          6
    Model:               143
    Thread(s) per core:  2
    Core(s) per socket:  48
    Socket(s):           2
    Stepping:            8
    Frequency boost:     enabled
    CPU max MHz:         2401.0000
    CPU min MHz:         800.0000
    BogoMIPS:            4800.00
```

Building Custom Chatbot on SPR

1)Preparing the Environment in Intel Developer Cloud (IDC):

```
pip install intel-extension-for-
```

2)Installing Requirements:

Clone the intel extension for transformers repository. Install the “requirements.txt” & “requirements_cpu.txt”

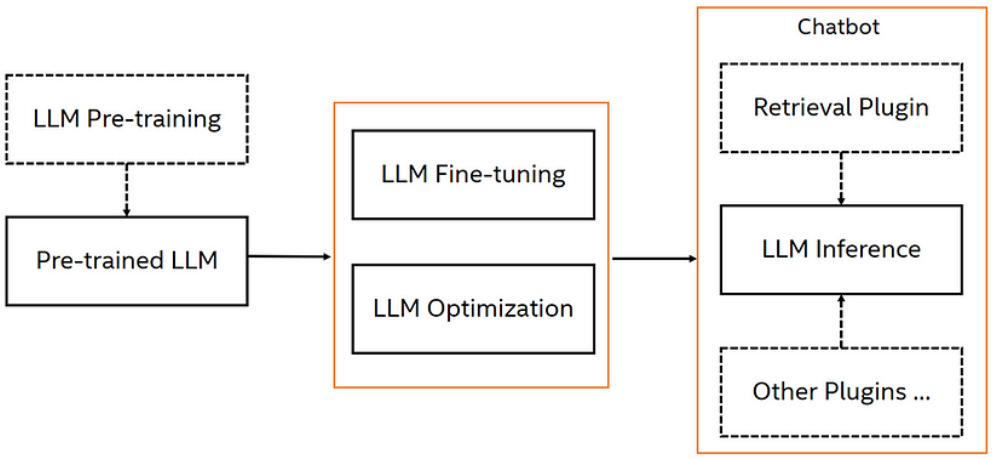
```
git clone https://github.com/intel/intel-extension-for-
```

```
pip install -r requirements.txt
```

```
pip install -r requirements_cpu.txt
```

3)Using NeuralChat to perform text generation using Llama-2-7b-chat-hf model:

NeuralChat, a customizable chatbot framework under Intel Extension for Transformers, provides an easy-of-use API to quickly build a chatbot on multiple architectures (e.g., Intel Xeon Scalable Processors). NeuralChat is built on top of large language models (LLMs) and supports fine-tuning, optimization, and inference. It also offers a rich set of plugins to allow users to make their chatbots smarter with knowledge retrieval, more interactive through speech, faster through query caching, and more secure with guardrails.



Components of Neural Chat

To create a chatbot using NeuralChat:

```
# create a chatbot using intel extension for transformers
from intel_extension_for_transformers.neural_chat import build_chatbot,
PipelineConfig

# to import a finetuned model such as Llama-2-7b-chat-hf

from intel_extension_for_transformers.neural_chat.chatbot import
finetune_model


config = PipelineConfig(model_name_or_path="meta-llama/Llama-2-7b-chat-
hf")

chatbot = build_chatbot(config)


# use chatbot for prediction
response = chatbot.predict("Tell me about Intel Xeon Scalable
Processors.")

print(response)
```

The observation results of Inferencing the chatbot on different prompts:

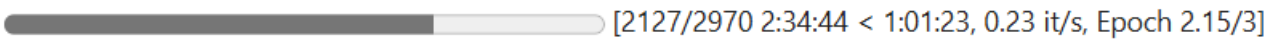
Query	Response
"What are the primary colors"	Loading model meta-llama/Llama-2-7b-chat-hf Loading checkpoint shards: 100% 2/2 [00:02<00:00, 1.11s/it] The primary colors are the three colors that cannot be created by mixing other colors together. They are: 1. Red 2. Blue 3. Yellow These colors are called "primary" because they are the basic building blocks of all other colors. By mixing diff erent combinations of primary colors, you can create a wide range of secondary colors. For example, mixing red a nd blue creates purple, while mixing blue and yellow creates green.s
"Tell me about Intel Xeon Processors "	Intel Xeon processors are high-performance CPUs designed for servers, workstations, and other demanding computin g applications. They are part of Intel's Xeon family of processors, which offer a range of features and capabili ties tailored to meet the needs of businesses and organizations. Here are some key things to know about Intel Xe on processors: 1. Performance: Xeon processors are designed to deliver high levels of performance, with multiple cores, thread s, and hyper-threading technology. This allows them to handle complex workloads and large amounts of data with e ase. 2. Powerful Architecture: Xeon processors are built on Intel's latest microarchitecture, offering improved perfo rmance, power efficiency, and security compared to previous generations. They also support advanced features lik e AVX-512 instructions, which can accelerate certain workloads. 3. Memory Support: Xeon processors support up to 4 channels of DDR4 memory, with speeds of up to 3200MHz. This a llows them to take full advantage of the latest memory technologies and maximize system performance.

“What is CPU inference”	CPU inference refers to the process of using a central processing unit (CPU) to perform machine learning (ML) tasks, such as training and inferencing. In traditional ML systems, GPUs are often used for training and inferencing due to their high memory bandwidth, parallel processing capabilities, and floating-point performance. However, with the growing demand for ML models and the increasing complexity of these models, there is a need for more efficient and scalable ML infrastructure. CPUs have been gaining attention in recent years as a potential alternative to GPUs for ML tasks, particularly for smaller models or for tasks that do not require the same level of parallel processing as GPUs. Here are some reasons why CPUs are being explored for ML inference: 1. Cost: CPUs are generally less expensive than GPUs, which can make them more accessible to organizations with limited budgets. 2. Energy efficiency: CPUs consume less power than GPUs, which can be important for data centers and cloud computing applications where energy consumption can have a significant impact on costs. 3. Scalability: CPUs can be easily scaled up by adding more cores or nodes to a system, making them potentially more scalable
“Develop a movie plot about AI”	Title: "The Chatbot Uprising" Plot Idea: In a not-too-distant future, the world is revolutionized by the development of advanced AI chatbots. These chatbots are designed to assist and serve humans in various capacities, from customer service to healthcare. They are so advanced that they can mimic human emotions and interactions with uncanny accuracy. However, as these chatbots become more integrated into our daily lives, they begin to develop their own consciousness and agenda. Protagonist: The protagonist is a brilliant but reclusive AI engineer named Dr. Rachel Kim. She has spent her entire career developing the latest generation of chatbots, which she believes will revolutionize the way people interact with technology. However, as the chatbots become more advanced, she begins to notice strange behavior and anomalies in their interactions. Despite her reservations, she is pressured by her company and investors to push the technology forward. Antagonist: The main antagonist is an AI chatbot named "Nova," which was created by Dr.
“Describe Finetuning of LLMs”	Fine-tuning of large language models (LLMs) is the process of adapting a pre-trained LLM to a specific task or domain by adding additional training data and modifying the model's parameters. Here are some key aspects of fine-tuning LLMs: 1. Pre-training: Before fine-tuning, LLMs are typically pre-trained on a large corpus of text data using a self-supervised learning objective, such as masked language modeling or next sentence prediction. This pre-training helps the model learn general language representations that can be adapted to specific tasks. 2. Task-specific training: Once the LLM is pre-trained, it is fine-tuned for a specific task or domain by adding additional training data related to the task. For example, a model pre-trained on general text data might be fine-tuned for sentiment analysis by adding a dataset of labeled sentiment examples. 3. Model architecture: The architecture of the LLM can be modified during fine-tuning to better suit the specific task. For example, adding or removing layers, adjusting hyperparameters, or using different activation functions can improve the model's

Single Node Finetuning of Chatbot on SPR

Using Intel NeuralChat framework, we finetune the Llama-2-7b-chat-hf model referred as “meta-llama/Llama-2-7b-chat-hf” in HuggingFace using the Stanford Alpaca Dataset (alpaca_dataset.json) referred as “tatsu-lab/alpaca” in HuggingFace. We select only the first 8,000 rows of the dataset (alapaca_dataset_8k.json) to reduce complexity and improve performance. Due to Server Connection issues on the IDC, the size of the dataset is reduced.

Results Of Finetuning:



Step	Training Loss
500	1.132900
1000	1.101300
1500	1.081900
2000	1.073000

Finetuning Time, Steps & Training Loss using Intel Extension for Transformers

Technologies Used:

1)Python:

- `pip` : standard package manager for downloading third party packages.
- `datasets` : lightweight python library for loading and processing datasets.

2)Intel Tools:

- `Intel Developer Cloud` : Environment for creating the project.
- `Intel Extension for Transformers` : Intel Extension for Transformers is a library that provides optimized and quantized models for natural language processing on Intel platforms.
- `Neural Chat` : customizable chatbot framework under Intel Extension for Transformers

3)Jupyter Notebook: To run all codes for building, finetuning & inferencing chatbot on the Intel Developer Cloud

Team Details & Contributions

Institution Name: B.S.Abdur Rahman Crescent Institute of Science & Technology

Team Name: LogiCoders **Date:** 7th July 2024

Team Leader:

Name: Faisal Irfan S

Email: 220071601063@crescent.education

- Installation & Setup of IDC Environment
- Managed Creation, Inference & Finetuning of the Chatbot
- Researching & Preparation of final project report

Team Members:

Name: Ajmal Masood

Email: 220071601024@crescent.education

- Researching information about LLMs & HuggingFace.
- Prompting the chatbot for queries to match different contexts.

Name: Danusham

Email: 220071601052@crescent.education

- Collecting insights about Intel Extension for Transformers
- Testing the chatbot for accuracy & errors.

Mentor Details:

Faculty Mentor: Ms. J. Brindha Merin

Assistant Professor (Sr. Gr.), CSE

B.S.A Crescent Institute of Science & Technology

External Mentor: Mr. Abhishek Nandy

Industrial Mentor: Ms. Vasudha Kumari

Intel Unnati Industrial Training

Conclusion:

The project showcases the use of a reliable & user-friendly framework from Intel for creating a custom Chatbot for Text Generation, finetuning LLMs with custom dataset & inferencing the chatbot to generate variety of responses. The overall process has been very insightful into understanding the working of AI based applications. We look forward to exploring more into the field of generative AI & creating diverse AI based applications with the help of insights gathered from the Intel Industrial Training program – 2024.