COMPARATIVE ANALYSIS OF OPEN-SOURCE FOUNDATION LARGE LANGUAGE MODELS FOR FINANCE DOMAIN THROUGH IMPLEMENTATION

VIJAYSHANKAR SUBRAMANIAN

Final Thesis Report

DECEMBER 2023

ABSTRACT

Commercial Large Language Models (LLMs) like ChatGPT has disrupted the field of Natural Language Processing (NLP) by showcasing outstanding performance in multiple NLP tasks. However, there are data security and privacy concerns around using commercial LLMs since they require the user queries to be sent to their server for processing. Open-source LLMs can help address this challenge since they can be hosted in enterprises' own infrastructure. For NLP tasks based on a specific domain (like finance), research shows that LLMs fine-tuned on domain specific data improves the performance of LLMs. There are many open-source LLMs available. Therefore, it is necessary to fine-tune the LLMs and perform comparative analysis to identify the strengths and weaknesses of a LLM for a specific domain like finance. State-of-the-art research on LLMs for finance domain is focussed on training or fine-tuning a specific LLM using finance domain data. To the best of author's knowledge, there are no other studies which fine-tune multiple open-source LLMs on financial data and provide comparative analysis of LLMs for finance domain. Therefore, this study addresses this research gap by considering three state-of-the-art open-source foundation LLMs (LLaMA-2, Mistral and Falcon) and fine-tuning all of them on financial data and performing comprehensive comparative analysis of LLMs using appropriate evaluation metrics. Elaborate literature review was performed to select the key NLP tasks (sentiment analysis and question-answering) relevant to finance domain. End-to-end programming framework was developed to fine-tune the LLMs and perform evaluation. The LLMs were fine-tuned and evaluated, and results of this study reveal that Mistral is the best performing LLM based on comparative analysis of LLMs across NLP tasks. Results also reveal all three LLMs show outstanding performance on sentiment analysis task, however, underperform on question-answering task. The results and insights from this study will enable the finance domain stakeholders to choose appropriate LLM for their use case.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AGI….......................	Artificial General Intelligence
ANN…......................	Artificial Neural Networks
BERT ….....................	Bidirectional Encoder Representations from Transformers
BLOOM…...................	BigScience Large Open-science Open-access Multilingual Language Model
BOW…......................	Bag-of-Words
BPE….......................	Byte Pair Encoding
CBOW…....................	Continuous Bag of Words
CNN…......................	Convolutional Neural Networks
DNN…......................	Deep Neural Networks
DT….........................	Decision Trees
ELECTRA…...............	Efficiently Learning an Encoder that Classifies Token Replacements Accurately
EM…........................	Exact Match
FLANG….................	Financial Language model
FinBERT….................	Finance Bidirectional Encoder Representations from Transformers
GPT….......................	Generative Pre-Trained Transformer
GPU…......................	Graphical Processing Unit
GQA…......................	Grouped Query Attention
ICL…........................	In-Context Learning
KNN…......................	K Nearest Neighbours
LaMDA ….................	Language Model for Dialogue Applications
LLM…......................	Large Language Model
LoRA….....................	Low Rank Adaptation
LSTM …...................	Long Short-Term Memory
MDR ….....................	MacroData Refinement
ML…........................	Machine Learning
MLM…....................	Masked Language Models
MPT….....................	MosaicML Pretrained Transformer

MRC…......................	Machine Reading Comprehension
NB….........................	Naïve Bayes
NER….......................	Named Entity Recognition
NLP….......................	Natural Language Processing
OPT….......................	Open Pre-trained Transformers
PaLM …....................	Pathways Language Model
PEFT….....................	Parameter efficient fine tuning
PII….........................	Personally Identifiable Information
POS….......................	Parts-of-Speech
QLoRA…..................	Quantized Low Rank Adaptation
RF….........................	Random Forests
RLHF …....................	Reinforcement Learning from Human Feedback
RNN…......................	Recurrent Neural Networks
RoBERTa…...............	Robustly Optimized BERT Approach
SVM…......................	Support Vector Machines
SWA…......................	Sliding Window Attention
TAT-QA..................	Tabular And Textual dataset for Question Answering
TF-IDF…...................	Term Frequency Inverse Document Frequency
TII….........................	Technological Innovation Institute

CHAPTER 1

INTRODUCTION

1.1 Background

LLM is a deep learning model which has been trained on vast amounts of text data from multiple sources like websites, books and various other forms of text content. LLMs have showcased state-of-the-art performance in variety of NLP tasks (Brown et al., 2020b). GPT (Generative Pre-trained Transformer) models are LLMs which are based on transformer neural network architecture (Vaswani et al., 2017). GPT-1 (Radford et al., 2018) was the first model in GPT series of LLMs and it was built by using a hybrid approach of unsupervised pre-training and supervised fine-tuning. This model laid the foundation for GPT series models and established the underlying principle to model natural language text which is predicting the next word. GPT-2 (Radford et al., 2019) had similar architecture as GPT-1 but number of parameters were increased to 1.5 billion and model was trained on larger dataset.

Introduction of GPT-3 (Brown et al., 2020a) by OpenAI was a major turning point in development of LLMs. GPT-3 increased number of parameters to 175 billion and introduces in-context learning (ICL). ICL can instruct LLMs to understand tasks in the form of natural language text. GPT-3 shows excellent performance in a variety of NLP tasks like answering user questions, translation, perform summarisation and generating text. GPT-3 also performs well on tasks that require reasoning abilities and domain adaptation. GPT-3 has been used as foundation model to develop even more capable LLMs. However, GPT-3 had few limitations around tasks like completing code and solving math problems. Therefore, codex (Chen et al., 2021a), a GPT model fine-tuned on GitHub code was released by OpenAI. Codex was used to build GitHub co-pilot which can take a programming problem described in natural language as input and generate solution code for the same.

GPT-3.5 was derived from GPT-3 and changes were made on top of GPT-3 to remove toxic output and align the output with user’s intention. ChatGPT, a GPT model optimized for dialogue was built based on GPT-3.5. ChatGPT became one of fastest growing consumer applications in history with around 100 million users monthly (Wu et al., 2023b). Apart from general LLMs, there are domain specific LLMs like BioGPT (for medicine) and BloombergGPT (for finance) which were trained on vast amounts of data in specific domain.

The GPT-3.5 model developed by OpenAI produced state-of-the-art results. However, GPT-3.5 is a commercial LLM and hence code and dataset used to train the model is not available in public domain. Commercial LLMs require the user queries and data to be sent to their server for processing, in addition to this, some LLMs like ChatGPT save user prompts to train and improve its models. It is also possible some users may inadvertently enter sensitive or personally identifiable information (PII) when specifying the query and this will be sent to servers of commercial LLMs.

Enterprises are skeptical to use commercial LLMs due to above mentioned privacy and security concerns. Open-source LLMs can address the challenges that comes with using commercial LLMs for enterprises. Enterprises can use open-source LLMs to deploy these models on their own infrastructure (on-premises or private cloud environment) ensuring sensitive information remains within the enterprise network and thereby reducing the risk of data breaches. Apart from improved security, another major advantage is that there is no additional cost involved in using open-source LLMs since it is freely available and hence very useful for enterprises with limited budget.

LLMs which are trained from scratch using massive amounts of text are known as foundation models (Zhou et al., 2023). LLaMA-1, LLaMA-2, Falcon, Mistral, OPT (Open Pre-trained Transformers) and BLOOM (BigScience Large Open-science Open-access Multilingual Language Model) are some of open-source foundation LLMs. Research (Gururangan et al., 2020) shows that language models fine-tuned on data for a specific domain performs better on specific tasks in that domain compared to language models which have not been fine-tuned. There are many open-source LLMs available and hence comparative analysis is required to identify the LLM which is best suited for a specific domain like finance. The field of open-source LLMs is relatively new and existing studies for finance domain are related to training or fine-tuning a specific LLM using finance domain data (Wu et al., 2023). To the best of author's knowledge, there are no other studies which fine-tune multiple open-source LLMs on financial data and present comparative analysis on the same. Therefore, this study aims to bridge this research gap by considering three top open-source foundation LLMs – LLaMA-2, Falcon and Mistral and fine-tuning all of them on financial data and evaluating them on important NLP tasks (question answering and sentiment analysis) using appropriate evaluation metrics and provide a thorough and comprehensive comparative analysis. The results of this study can help stakeholders in finance domain (both functional and technical capacities) to choose the appropriate open-source LLM for their enterprise use case.

1.2 Problem Statement

NLP has been a key driver in advancement of finance technology by enabling multiple capabilities from stock price forecasting to advanced financial analytics. Introduction of LLMs has been the most exciting and promising development in field of NLP and they have shown remarkable performance in variety of NLP tasks (Brown et al., 2020b). While the performance of LLMs in general NLP tasks has been outstanding, studies have been shown that language models fine-tuned on domain data perform better on domain specific tasks. FinBERT (Finance Bidirectional Encoder Representations from Transformers) is one of the first pre-trained language models which is fine-tuned using financial data. The FinBERT (Yang et al., 2020) takes BERT (Bidirectional Encoder Representations from Transformers) an existing pre-trained language model as base and performs fine-tuning on vast amount of financial communication data including earning call transcripts, analyst and corporate reports. Evaluation was done by comparing the accuracy of FinBERT with generic BERT model on three financial sentiment detection datasets. This study reports that FinBERT was able to outperform BERT.

Based on Financial Language Model (FLANG), (Shah et al., 2022) introduces two pre-trained models fine-tuned on financial data – FLANG-BERT and FLANG-ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately). The base models for these two models are BERT and ELECTRA respectively (Clark et al., 2020). FLANG aims to improve upon FinBERT by employing preferential token masking in masked language models during training. The main objective of masked language models (MLM) is to predict the masked token in a particular sentence. Usually, tokens are masked with equal probability. However, in preferential token masking, preference is given to certain tokens. FLANG models are reported to have outperformed generic BERT, ELECTRA and FinBERT in different tasks like question answering, sentiment analysis and named entity recognition by this study.

Introduction of BloombergGPT (Wu et al., 2023) marked a major milestone in development of LLMs for finance. BloombergGPT is a 50 billion parameter LLM which has been trained on a mix of vast amounts of Bloomberg finance data and public data. BloombergGPT is evaluated on both finance related and general tasks, both Bloomberg datasets and publicly available finance datasets are utilized for evaluation on finance related tasks. This study reports that BloombergGPT was able to provide superior performance on finance related tasks compared to other models, whereas on general tasks, it is reported to be on par with other models. Despite the outstanding performance BloombergGPT on finance tasks, there are some limitations which makes it difficult for finance enterprises to adopt it. Data security is very important for banking and other financial institutions since they hold large amount of sensitive data (Alghazo et al., 2017). BloombergGPT is a commercial LLM which requires the user queries to be sent to its server for processing and hence this is one of the limitations in its adoption by finance enterprises. Cost involved in using BloombergGPT is another limitation. Open-source LLMs offer a viable alternative to BloombergGPT since enterprises can host them in their own infrastructure and there is no additional cost involved in using open-source LLMs. Finance enterprises can use one of many open-source LLMs by fine tuning them on finance datasets to improve the performance on finance related tasks.

There are multiple open-source foundation LLMs available that can be adopted by finance domain enterprises. However, there are no comparative studies available which can help finance organisations to take the decision on adopting best available open-source foundation LLM. Therefore, this work aims to present comparative analysis on the same. This work intends to adopt three top open-source LLM’s i.e., LLaMA-2, Falcon, and Mistral. The work aims to fine-tune all of them on financial data and evaluating them on important NLP tasks using appropriate evaluation metrics and provide a thorough and comprehensive comparative analysis.

Instruction tuning will be used to fine-tune the LLMs. This is a type of supervised fine-tuning, in which, a collection of instructions and corresponding examples of how to follow the instructions are used to train the LLMs. Instruction tuning datasets will be created by overlaying the instructions (prompts) on the datasets used for this study, these instruction tuning datasets will be used to fine-tune the LLMs. Instruction tuning has been shown to improve performance of LLMs (Wei et al., 2021).

1.3 Aim and Objectives

The main aim of this research is to perform a thorough and comprehensive comparative analysis of open-source LLMs for finance domain by fine-tuning on finance data and evaluating the LLMs on multiple NLP tasks. The research objectives based on above mentioned aim are as follows:

- To create instruction tuning datasets by overlaying the instructions (prompts) on train dataset.
- To build comprehensive end-to-end programming framework to fine-tune the LLMs.
- To fine-tune the LLMs on instruction tuning datasets.
- To evaluate the fine-tuned models on NLP tasks relevant to finance domain.

1.4 Significance of the Study

This work is of significance to technical and functional stakeholders in finance who are interested in choosing the best suited open-source LLM as per the requirements for their use case. There are multiple open-source LLMs available and comparative analysis is required to identify the best suited LLM. The results and insights from comparative analysis of open-source LLMs done in this study will provide them with necessary details and metrics required to make the right decision.

1.5 Scope of the Study

This study considers three top foundation open-source LLMs – LLaMA-2, Mistral, and Falcon. These LLMs are considered since they are among the best open-source foundation LLMs. There are other open-source LLMs which are not in the scope of this study and can form the basis for future work.

Evaluation of LLMs is done on question-answering and sentiment analysis NLP tasks. These NLP tasks have been selected based on two factors – first is importance and relevance of these tasks for financial domain and second is timeframe allocated for this study. Details of reasons for choosing these tasks are given in literature review chapter (section 2.4.1 and 2.4.2). There are other NLP tasks on which evaluation of LLM could be done but those are out of scope of this study and can be considered for future work. LLMs typically have models of varying parameter sizes. Falcon LLM has models ranging from 7 billion parameters to 180 billion parameters. LLaMA-2 has models ranging from 7 billion parameters to 70 billion parameters.

Mistral LLM has only one variant with 7 billion parameters. Due to the large size of the models, high end GPUs (Graphical Processing Units) are required to train and evaluate them on large datasets. The high end GPUs are provided only by paid cloud-based platforms like Google Colab Pro Plus and hence this platform has been used in this research for fine-tuning and evaluating the LLMs. Due to the compute resources (GPUs) and financial resources (to use the cloud platform) required for this research, 7 billion parameter variants of LLaMA-2, Falcon and Mistral LLMs will be considered for this research. The other variants of LLaMA-2 and Falcon are not in the scope of this study.

1.6 Structure of the Study

The structure of the study includes six chapters as summarized below. The report begins with Chapter 1 (Introduction). This chapter provides background and problem statement for this research. It also defines aims and objectives for the proposed work based on literature review (presented in Chapter 2). Chapter 2 (Literature Review) presents comprehensive literature review on language models used in finance domain. This chapter also discusses in detail about the literature on key NLP tasks used to evaluate the LLMs and its importance and relevance to finance domain. Literature review on LLMs used in this study has also been provided in this chapter. Chapter 3 (Research Methodology) outlines the steps involved, various methods, and techniques used in this research. This chapter discusses about transformers since it forms the foundational component in architecture of all the LLMs used in this study. Fine-tuning is a crucial component of this study. Hence, LoRA optimisation technique used to save computational resources while fine-tuning the LLMs are also discussed in this chapter. Detailed overview of LLMs and evaluation metrics used in this work has also been discussed. Chapter 4 (Analysis and Design) delves into the details of analysis and implementation of this study. This chapter discusses about the crucial hyperparameters used in this study. This chapter also provides the details about the platform, various libraries and packages used to develop the programming framework used in this study. Chapter 5 (Results and Discussion) provides a thorough and comprehensive comparative analysis using the evaluation results for LLMs used in this study. The interpretation and relevance of these results has also been provided in this section. Chapter 6 (Conclusion) provides summary of the findings of this study with respect to the stated objectives. This chapter also discusses about the limitations and future recommendations.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The main objective of this study is to fine-tune the open-source LLMs on finance domain data and evaluate the LLMs and present comparative analysis. Therefore, this chapter presents comprehensive literature review on related topics. LLMs are a sub-field in the broader NLP domain and hence before getting into the details of LLMs, section 2.2 presents overview of NLP models leading up to the development of LLMs. Section 2.3 focusses on the literature on LLMs. It includes discussion about evolution of GPT series of models and datasets used to train these models. Further, this section gives an overview of literature on popular open-source LLMs like LLaMA, Mistral, Falcon, Alpaca, Vicuna and Instruct-GPT. The rationale for selecting three open-source LLMs (LLaMA-2, Mistral, and Falcon) for evaluation and comparative analysis in this research are also discussed in this section. Section 2.4 presents literature review about the NLP tasks which will be used for evaluating the LLMs in this work. The reasoning behind selecting question-answering and sentiment analysis NLP tasks to evaluate the LLMs are also provided in this section. This section also presents literature on evaluation metrics which will be used to evaluate the LLMs in this study. Finally, section 2.5 presents discussion about research gaps identified based on which aims and objectives of this thesis are defined.

2.2 Overview of NLP

Humans use natural language to communicate with each other, however, sentences and words spoken by humans cannot be understood by computers. NLP is a set of techniques developed based on Machine Learning (ML) which enables computers to understand and comprehend natural languages. The concepts of Artificial Intelligence and Linguistics are used to develop NLP techniques and models.

2.2.1 Data Pre-processing in NLP

The text data derived from different sources is generally unstructured and contains lots of unimportant information. Data pre-processing involves using various techniques to clean up the text and convert it into a consistent format which can be given as input to ML model. (Al-Hawari and Barham, 2021) compares the accuracy of models trained on data that was cleaned and pre-processed with models trained on dataset which was not pre-processed. The experiments conducted by this study shows that there was significant increase in accuracy when the dataset is cleaned and pre-processed before training the model. This shows the importance of data pre-processing in the context of NLP. (Tabassum and Patil, 2020) presented a survey on data pre-processing techniques in NLP. This study shows the most frequently used data pre-processing techniques are changing text to lower case, stop words removal, removing punctuations, tokenisation, parts-of-speech (POS) tagging, stemming and lemmatization.

2.2.2 Feature Extraction

ML models require numerical inputs and hence natural language text cannot be given as input. Feature extraction involves taking text data and converting it into feature vector which can be given as input to the ML model for training and prediction. Bag-of-words (BOW), Term frequency – inverse document frequency (TF-IDF) and word2vec are some of the feature extraction techniques used in NLP (Tabassum and Patil, 2020). The major drawback with both BOW and TF-IDF is that it creates sparse vector representation with lots of features where most of words are represented with zero. This can be resolved by using word embedding techniques like word2vec.

(Mikolov et al., 2013) introduces word2vec embedding technique and mentions that techniques like BOW and TF-IDF treat words as independent units and do not consider the similarity between the words when creating the feature vector representation. The study states that main goal of word2vec is to create dense vector representation of words while also considering the similarity between words. As per the study, major advantage of word2vec algorithm is that it creates dense vectors resulting in faster computations and it can capture the meaning of the word along with similarity with surrounding words. The study mentions that shallow neural network is trained on input text data to create the word embeddings. As per the study, there are two modes in word2vec – continuous bag-of-words (CBOW) and skip-gram. CBOW predicts the central words based on the surrounding words. Skip-gram model predicts the surrounding text based on the central word. One of the drawbacks of word2vec is that it assigns random vector values to words which was not part of the training corpus.

2.2.3 NLP Models

Raw text data is pre-processed, and numerical feature vectors are constructed using feature extraction techniques. Feature vectors are given as input to ML models. (Otter et al., 2021) discusses about various ML models used in the field of NLP. This study states that conventional ML models such as support vector machines (SVM), Random Forests (RF), Decision Trees (DT), Naïve Bayes (NB) and K-Nearest Neighbours (KNN) were used in the past for NLP problems. As per this study, there has been complete transformation in the past few years, where approaches using conventional ML models have been replaced by models based on neural networks. This study states that neural networks comprise of neurons (which are interconnected nodes) receiving some number of inputs and producing an output. Each node in the network calculates a weighted sum of input values which they receive from input nodes and apply some transformation function on weighted sum of input values to produce the output. As per the study, there are two factors which determine the type of neural network – way nodes are connected to each other and number of layers in the network. Basic neural networks have layers of nodes arranged in a sequential manner in a way that nodes receive input only from nodes in previous layers. This type of neural network is known as Artificial Neural Network (ANN). On the other hand, neural networks with many layers are known as Deep Neural Networks (DNN) (Schmidhuber, 2015).

(Li et al., 2022b) presented a detail literature review of Convolutional Neural Networks (CNNs). This study states that CNN is a type of neural network that extracts features from data using convolutional structures. This study mentions that CNN does not extract features manually unlike traditional feature extraction methods. The study states that CNN is made up of input layer, convolution layer, pooling layer and fully connected layer. As per the study, inputs are provided to CNN using the input layer and filters are applied to the features using convolution functions in the convolution layer to create feature maps. Pooling layer is applied after convolution layer and main purpose of this layer is to reduce the dimensions of feature map. The flattened feature vector from pooling layer is given as input to fully connected layer. The study states that flattened feature vector is an ANN where all the nodes in a particular hidden layer are fully connected to nodes in next hidden layer. Activation function is applied in the fully connected layer to produce the final output. As per this study, the main advantages of CNN are – local connections, weight sharing and down sampling dimension reduction. The study states that neurons in CNN are not connected to all neurons in previous layer, instead they are connected only to small number of neurons, and this is known as local connections. This can help with reducing parameters and speeding up convergence. A group of connections can also share the same weights, thereby, reducing the number of parameters further, this is known as weight sharing. The study also states that pooling layer which is added to CNN uses down sampling to consolidate the features learned and thereby, resulting in dimension reduction and this is known as down sampling dimension reduction.

(Tarwani and Edem, 2017) performs a survey on Recurrent Neural Networks (RNNs) used in NLP. This study states that RNNs are a type of neural networks which not only considers the current input but also the previous inputs when producing the output. This study states that RNNs have memory of previous input words and hence well suited for NLP applications which require prediction of a word based on previous input words. As per this study, sequence labelling and prediction tasks like machine translation and language modelling are some of the major NLP applications using RNN. The study states that there are instances where it is useful to consider the words before and after the word for which prediction needs to be done. This can be achieved using bi-directional RNN where two unidirectional RNN layers (one predicting the current word based on previous words and another predicting the current word based on following words) are combined. As per the study, the main disadvantage of basic RNNs is that it has difficulty in learning about words that are many words far away from the current word. There are instances where prediction of current word may be dependent of words far away and basic RNNs have difficulty in making the correct prediction in such cases and this is known as vanishing gradient problem (Hong Hui Tan and King Hann Lim, 2019). This can be overcome by using LSTM (Long Short-Term Memory) (Greff et al., 2017). The study states that LSTM is a type of RNN where components are introduced to determine the proportion of long-term memory (importance of words far back in a sentence in predicting the current word) and short-term memory (importance of words nearer to word being predicted) to be retained to make accurate predictions. This study also mentions that this type of architecture can enable crucial information to be retained and unnecessary information to be ignored by RNN.

DNNs (neural networks with multiple hidden layers) have shown excellent performance on multiple tasks like speech recognition, computer vision and pattern recognition (Liu et al., 2017). As per the study, the major drawback of DNNs is that it can only be used in tasks where input and output sequences can be encoded into feature vectors of fixed dimensionality. This limitation has a significant impact on tasks like machine translation where sequence length is not fixed (input sentences and output sentences may be of different length). This limitation can be mitigated using encoder decoder neural network model (Sutskever et al., 2014). This study mentions that encoder component of this model takes input sequences of different lengths and converts it into a fixed length vector. Decoder component of the model is used to regenerate the original input from the encoded feature vector and generate the desired output. This study mentions that LSTM models are used to build the encoder and decoder components. The study also reports that this model achieves good performance in machine translation task.

(Bahdanau et al., 2014) aims to improve upon encoder decoder models by introducing the concept of attention. The study states that one of the issues with encoder decoder models is that encoder component needs to compress all the required information into fixed length feature vector. As per the study, performance of basic encoder decoder model degrades when there is an increase in the length of input sentences This study proposes an approach where decoder focusses on the most important parts of the fixed length vector produced by the encoder, and this is known as attention. This study reports that this model can outperform basic encoder decoder model on machine translation and performs well on long sentences. This study also mentions that encoder decoder models with attention mechanism are built using RNNs. The major disadvantage of recurrent models like RNN is that training must be done sequentially since the previous input must be considered. Parallelization which allows more faster and efficient computing cannot be used to due to sequential nature of recurrent models. (Vaswani et al., 2017) introduced transformer architecture which removes recurrent components and uses attention mechanism to capture the relevant information. This architecture allows for parallelization during training since recurrent components are removed and hence transformers can be trained on large text corpora efficiently. The most important application of transformers is that it forms the basic building block for LLMs.

2.3 Large Language Models (LLMs)

(Min et al., 2024) defined language modelling as a task where model predicts a particular word based on the context (words before or after the word being predicted). Models which are trained using this task are known as language models. Language models based on transformer architecture and containing billions of parameters and trained on massive amounts of text data are known as LLMs (Zhao et al., 2023a). Studies (Wei et al., 2022) have shown that increase in scale of language model (number of parameters and size of data on which model is trained) leads to better performance in multiple tasks. The abilities that are present in large models but not present in small models are known as emergent abilities. This is one of the important factors which distinguish LLMs from other pre-trained language models of smaller size.

2.3.1 Generative Pre-Trained Transformers (GPT)

GPT series of models (GPT-1, GPT-2, GPT-3, and GPT-4) developed by OpenAI are based on decoder only transformer architecture (Zhao et al., 2023a). This study attributes the success of GPT models to two factors – transformer architecture capable of accurately predicting the next word and ability to scale up the size of language models. (Radford et al., 2018) introduces GPT-1. This study mentions that large amount of unlabelled text data is available but labelled data is scarce and this poses a challenge from training perspective. To mitigate this problem, this study proposes generative pre-training of a language model on unlabelled text data followed by fine-tuning on specific task. GPT-1 is evaluated on four different language understanding tasks and study reports that performance improves upon the state-of-the-art results.

GPT-2 (Radford et al., 2019) has architecture similar to GPT-1 but size of the model is increased to 1.5 billion parameters and model is trained on a large dataset of webpages (WebText). This study reports impressive results on 7 out of 8 language modelling datasets used for evaluation. GPT-3 (Brown et al., 2020a) further scales up the number of parameters to 175 billion and introduces in-context learning. ICL is a process of providing few examples of user queries and corresponding answers along with actual user query. This is also known as few-shot prompting.

The introduction of ChatGPT by OpenAI marked a major milestone in the evolution of LLMs and garnered world-wide attention. ChatGPT surpassed 100 million users just two months after its launch (Wu et al., 2023b). (Ye et al., 2023) discusses about models which form the basis for ChatGPT. As per this study, OpenAI introduced code-davinci-002 for code generation tasks and this is the base model for GPT-3.5 series of models. OpenAI further improved this model using supervised fine-tuning and Reinforcement Learning from Human Feedback (RLHF) training techniques to create text-davinci-003. This model has improved ability to generate text and understand instructions. The text-davinci-003 model was further optimized for chat and this model was released as ChatGPT.

GPT-4 (OpenAI, 2023) further improves upon GPT-3 and displays human level performance on key professional and academic benchmarks. Another key improvement in GPT-4 includes that it can accept inputs of multiple types like images. (Bubeck et al., 2023) shows that GPT-4, apart from mastering the language, can solve problems in multiple domains like mathematics, medicine, coding and law without the need of any special method of prompting. This study mentions that due to GPT-4's extensive knowledge across domains, it could be viewed as early version of artificial general intelligence (AGI) system. As per the studies which introduced GPT series of models, they have performed well on variety of NLP tasks. However, GPT-3.5 and GPT-4 models are commercial LLMs and require the user queries to be sent to their server for processing. This could be a major issue for companies and enterprises which have sensitive data and do not prefer to send their data outside of their own servers (Bandari, 2023). Another, disadvantage of commercial LLMs is the cost associated with using them and this could be a problem for companies with limited budget. These issues can be mitigated by using open-source LLMs.

2.3.2 Open-Source LLMs

Open-source LLMs are free to use and can be used locally by companies without having to send data outside of their own server. Overview of important and popular LLMs are given in this section. Foundation models (Zhou et al., 2023) are LLMs which are trained from scratch using massive amounts of text. Fine-tuned models take foundation models as base and fine-tune on domain or task-specific data for performing tasks related to specific domains. LLaMA -1 (Touvron et al., 2023a) is one of most popular open-source foundation LLM released by Meta. There are four variants of this model in terms of number of parameters (in billions) – 65B, 30B, 13B and 7B. LLaMA-1 has been trained on variety on public data sources like common crawl, C4, GitHub, Wikipedia and StackExchange. The study also states that LLaMA-1 has been evaluated on multiple NLP tasks. The model was evaluated on eight common sense reasoning benchmark datasets and compared against three models - PaLM (Pathways Language Model), Chinchilla and GPT-3. It is reported by the study that LLaMA-1 (65B) was able to outperform Chinchilla on all benchmark datasets for this task. It is also reported by the study that LLaMA-1 (13B) was able to outperform GPT-3 on most benchmarks despite being ten times smaller. The study evaluated LLaMA-1 on closed-book question answering benchmark datasets and reported that LLaMA-1 (65B) model was able to achieve state-of-the-art performance. The study also evaluates LLaMA-1 on code generation tasks and reported that the model was able to outperform other general models like LaMDA (Language Model for Dialogue Applications) and PaLM.

Meta released LLaMA-2 (Touvron et al., 2023b) in an effort to improve upon LLaMA-1. The parameter sizes for LLaMA-2 ranges from 7B to 70B. The study states that much larger dataset (compared to LLaMA-1) which consists of texts from variety of public sources was used to train LLaMA-2. The study states that another major improvement for LLaMA-2 was increasing the context length to 4096 tokens (compared to context length of LLaMA-1 which was 2048 tokens). Study mentions that increase in context length enables the model to process more information. The study reports that evaluation of LLaMA-2 was done on multiple tasks and comparison was done against LLMs like MosaicML Pretrained Transformer (MPT) and Falcon. Commonsense-reasoning, reading comprehension, mathematical reasoning, and code-generation tasks were used for evaluation as per the study. It is reported by the study that LLaMA-2 model outperforms LLaMA-1. It is also mentioned by the study that LLaMA-2 7B and 30B models outperforms MPT models on all tasks except code generation. Study also reports that LLaMA-2 70B model outperformed Falcon and MPT models on all benchmarks. It is also reported by the study that LLaMA-2 7B and 34B models outperforms Falcon 7B and 40B models on all benchmarks. The study which introduced LLaMA-2 concludes that it is among the best open-source LLMs.

Mistral 7B (Jiang et al., 2023) is an open-source foundation LLM released by Mistral AI. This model has been trained on publicly available datasets. The study which introduces Mistral 7B states that unique feature of the models’ ability to achieve superior performance with relatively lesser number of parameters compared to other LLMs. As per the study, multiple NLP tasks like numerical reasoning, reading comprehension, commonsense reasoning and code generation have been used for evaluation. Mistral 7B has been evaluated against LLaMA-2 13B, LLaMA-2 7B and Code-LLaMA 7B models. As per the study, Mistral 7B was able to outperform LLaMA-2 on all benchmarks considered. The study also mentions that Mistral 7B was on par with Code-LLaMA model on code generation tasks. The evaluation results of the study show that Mistral 7B is one of best open-source LLMs.

Falcon (Penedo et al., 2023) is an open-source foundation LLM released by Technological Innovation Institute (TII) which has three variants based on number of parameters – 7B, 40B and 180B. The study mentions that unique factor about Falcon LLM is that it has been trained on web data alone. Generally, LLMs are trained on variety of data sources like technical documents, social media conversations, books and webpages. This study shows that powerful LLMs can be created by training using only high-quality web data which has been cleansed and pre-processed. The study states that eighteen benchmark datasets across multiple NLP tasks have been used for evaluation of the model. Falcon LLM has been compared against OPT series of models, GPT-3 models, GPT-Neo(X/J) models, Pythia and PaLM models. The evaluation results from the study shows that Falcon was able to outperform other LLMs. These results show that Falcon LLM is among the best open-source LLMs.

Alpaca (Taori et al., 2023) uses LLaMA-1 as the base foundation model and fine-tunes it using dataset consisting of 52k instructions. The instruction dataset has been generated using GPT-3.5 model. The study reports that Alpaca shows performance similar to that of GPT-3 model despite being comparatively small. Vicuna (Chiang et al., 2023) is another LLM which takes LLaMA-1 as foundation model and fine-tunes it using user shared conversations from ShareGPT (a google chrome extension which allows users share their ChatGPT conversations). Vicuna uses instruction tuning technique using above conversations dataset to fine-tune LLaMA-1. Instruct-GPT (Ouyang et al., 2022) which takes GPT-3 as the base model and fine-tunes on a manually collected dataset which consists of user prompts and completions which represents the desired model behaviour. The study reports that evaluation results show improvements in truthfulness and reduction in toxic output generation without compromising on performance.

The existing studies on LLMs for finance domain focus only on considering one base model and fine-tuning them on finance data. Literature review reveals that there are no existing studies which fine-tune multiple open-source foundation LLMs on finance data to provide comparative analysis through evaluation. Therefore, the main objective of this study is to fine-tune popular open-source foundation LLMs on finance datasets and present comparative analysis using thorough evaluation. Three open-source foundation LLMs – LLaMA-2, Falcon and Mistral 7B will be used in this research for comparative analysis. The evaluation conducted by studies (Jiang et al., 2023; Penedo et al., 2023; Touvron et al., 2023b) that introduced these LLMs have shown that these are among the best open-source foundation LLMs and hence will be used in this study for comparative analysis. This research only considers foundation LLMs since it involves fine-tuning on finance data, fine-tuned LLMs are already fine-tuned for a specific domain or tasks and hence not considered for this study.

2.3.3 Language Models in Finance Domain

The application of this study is in finance domain. Therefore, this section presents literature review on language models in finance domain. NLP has been a key driver in advancement of LLMs in finance domain and they have shown remarkable performance in variety of NLP tasks (Brown et al., 2020b). While the performnce of LLMs in general NLP tasks has been outstanding, studies have been shown that language models if fine-tuned on domain data perform better on domain specific tasks (Gururangan et al., 2020).

FinBERT (Finance Bidirectional Encoder Representations from Transformers Approach) is one of the first pre-trained language models to be fine-tuned on financial data. The study (Yang et al., 2020a) takes BERT (Devlin et al., 2019), an existing pre-trained language model and performs fine-tuning on vast amount of financial communication data including earning call transcripts, analyst and corporate reports. As per the study, evaluation has been done by comparing the accuracy of FinBERT with generic BERT model on three financial sentiment detection datasets and study reports that FinBERT was able to outperform BERT. FLANG (Shah et al., 2022) introduces two pre-trained models fine-tuned on financial data – FLANG-BERT and FLANG-ELECTRA. The base models for these two models are BERT and ELECTRA respectively(Clark et al., 2020). FLANG aims to improve upon FinBERT by employing preferential token masking in masked language models during training. The main objective of masked language models is to predict the masked token in a particular sentence. Usually, tokens are masked with equal probability. However, in preferential token masking, preference is given to certain tokens. FLANG models are reported to have outperformed generic BERT, ELECTRA and FinBERT in different tasks like question answering, sentiment analysis and named entity recognition by this study.

Introduction of BloombergGPT (Wu et al., 2023a) marked a major milestone in development of LLMs for finance. BloombergGPT is a 50 billion parameter LLM which has been trained on a mix of vast amounts of Bloomberg finance data and public data. Study states that base LLM for BloombergGPT is BLOOM. Study also mentions that BloombergGPT has been evaluated on both finance related and general tasks. As per the study, both Bloomberg datasets and publicly available finance datasets were utilized for evaluation on finance related tasks. This study reports that BloombergGPT was able to provide superior performance on finance related tasks compared to other models, whereas on general tasks, it is reported to be on par with other models. Despite the outstanding performance BloombergGPT on finance tasks, there are some limitations which makes it difficult for finance enterprises to adopt it. Data security is very important for banking and other financial institutions since they hold large amount of sensitive data. BloombergGPT is a commercial LLM which requires the user queries to be sent to its server for processing and hence this is one of the limitations in its adoption by finance enterprises. Cost involved in using BloombergGPT is another limitation. Open-source LLMs offer a viable alternative to BloombergGPT since enterprises can host them in their own infrastructure and there is no additional cost involved in using open-source LLMs. Finance enterprises can use one of many open-source LLMs by fine tuning them on finance datasets to improve the performance on finance related tasks.

Literature review reveals that there are many open-source language models such as, BERT, ELECTRA, LLaMA-1, LLaMA-2, Falcon, Mistral, MPT, OPT, BLOOM etc. These open-source LLMs can be adopted by finance domain enterprises. BERT and ELECTRA are older language models smaller in size in terms of parameters and come from pre-GPT era. The existing studies show that LLaMA-2, Falcon and Mistral are among the top open-source foundation LLMs and hence will be used in this research. The computational resources available and timeframe allocated for this study has also been taken in consideration when selecting the LLMs for this study. Literature review further indicates that there are no comparative studies available which can help finance organisations to take the decision on adopting best available open-source LLM.

2.3.4 Fine-Tuning LLMs

(Gururangan et al., 2020) shows that fine-tuning language models on domain specific data can lead to large gains in performance. Language models are generally pre-trained on variety of data sources and perform well on generalized tasks. This study considers datasets from four domains – computer science, biomedical, news and reviews. These datasets are used to fine-tune RoBERTa (Robustly Optimized BERT Approach) and evaluation is done on eight classification tasks (two for each domain). The result of this study shows that fine-tuning the models on domain specific data provides better results when compared to models which have not been fine-tuned. This shows the importance of fine-tuning of open-source foundation LLMs on finance domain datasets as part of this research.

Instruction tuning methodology is one of the methodologies used to fine-tune the LLMs and corresponding literature review is discussed in this section. Instruction tuning introduced by (Wei et al., 2021) is a method where tasks are described using instructions and these instructions are added to the data used for fine-tuning. Few-shot prompting involves providing LLMs few examples of a questions and answers along with original question for which output is required. Zero-shot prompting involves just providing the question as input to the LLM and generating output on the same. The study which introduces instruction tuning mentions that LLMs in general performs well under few-shot setting. This study mentions that GPT-3's zero shot performance was lesser compared to few-shot performance on tasks like question answering and reading comprehension. This study states that potential reason for this is that under zero shot setting the format of prompt was different from that of the format used during pre-training of the model. To mitigate this issue, this study introduces a technique of fine-tuning the model on multiple datasets expressed using natural language instructions. The datasets were grouped into different clusters based on their task types in this study. The models were fine-tuned using instruction tuning technique on multiple tasks like sentiment analysis, common sense reasoning and translation as part of evaluation in this study. The fine-tuned model was evaluated on a task like natural language inference. This experimental setup for this study ensures that the model has not seen any data related to the task on which it is evaluated. BERT model is considered for this study and results show that this technique substantially improves the performance under zero-shot setting. The study reports that fine-tuned model was able to outperform GPT-3 under zero shot setting for 20 of the 25 datasets being evaluated. This shows the need for instruction-tuning. In the proposed research, instructions will be added to train dataset as part of this research to create instruction-tuning datasets which will be used to fine-tune the LLMs on finance domain data.

LLMs typically have billions of parameters and performing full parameter fine-tuning on these models would mean updating these parameters and this can be very expensive from a computational perspective. Parameter efficient fine tuning (PEFT) provides an alternative in which only small proportion of parameters in the LLMs are fine-tuned. This significantly reduces the computational resources required for the fine-tuning process (Ding et al., 2023). Adapter tuning (Houlsby et al., 2019a) was among the initial methods of fine-tuning and this involves inserting adapter modules in the neural network architecture and only parameters for particular module will be fine-tuned. However, the main drawback with this technique is that inference latency was introduced since adapter layers were processed sequentially. LoRA

(Low Rank Adaptation) was introduced to overcome this limitation. The study which introduces LoRA (Hu et al., 2021) proposes a technique which involves performing low-rank decomposition of weight matrices of transformers which greatly reduces the number of trainable parameters. The study states that experiments conducted on GPT-3 175B model using this technique showed reduction in number of parameters by 10,000 times. The study also states that LoRA technique performs on-par with full parameter fine-tuning despite training fewer number of parameters. QLoRA (Quantized Low Rank Adaptation) is another technique which can be used to optimize the process of fine-tuning (Dettmers et al., 2023). As per this study, this technique involves optimizing the memory used to fine-tune the LLMs using the process of quantization. This study also states that this technique was able to reduce average memory requirement drastically without impacting performance. PEFT and QLoRA can be used together to fine-tune the LLMs in an optimal and efficient manner. This shows the effectiveness of LoRA and QLoRA fine-tuning techniques. Therefore, proposed work will use these techniques as part of this research to fine-tune LLMs on finance domain data.

2.4 LLM Tasks and Evaluation for Finance Domain

This section discusses about the NLP tasks and evaluation metrics used in existing literature on LLMs for finance domain. BloombergGPT (Wu et al., 2023a) is the biggest commercial LLM developed specifically for finance domain. The study which introduces BloombergGPT uses text classification (sentiment analysis), question-answering and named entity recognition (NER) tasks to evaluate the final model. This study uses sentiment analysis datasets from internal Bloomberg resources as test dataset to evaluate the performance of the model. Equity news sentiment dataset consists of news stories from Bloomberg company archive and has been annotated as ‘positive’, ‘negative’ or ‘neutral’ based on possibility of news causing the stock price to increase, decrease or not change respectively. Equity social media sentiment and equity transcript datasets are similar to equity news sentiment dataset but has social media and company transcript sentiment information instead of news. ES news sentiment and country news sentiment datasets contain data about news stories and corresponding sentiments about a country. ConvFinQA (Chen et al., 2022) dataset is used by the study which introduces BloombergGPT to evaluate the question-answering capabilities of the model for numerical reasoning tasks. This dataset contains data from earning reports of companies and hence that includes both text and tabular data. This task requires understanding of financial concepts and numerical reasoning to generate the correct answers. The study which introduces BloombergGPT uses NER dataset which consists of financial agreements with words annotated with relevant entity types (PER, LOC, ORG and MISC) to evaluate the model on NER task.

FLANG (Shah et al., 2022) fine tunes BERT and ELECTRA models on finance domain data using masked language modelling technique. This study uses five evaluation tasks – sentiment analysis, NER, question answering, news headline classification and structured boundary detection. FinBERT (Yang et al., 2020a) uses BERT as the foundation model and fine-tunes using financial data. This study uses sentiment analysis task to evaluate the FinBERT model and compare it against BERT to showcase its effectiveness. Financial phrase bank dataset (Malo et al., 2014) contains the sentiment labels for financial news and AnalystTone dataset (Huang et al., 2014) which contains sentiment labels for sentences in analyst reports is used as test dataset to evaluate the FinBERT model.

As per literature review, typically two to five tasks are used to evaluate the language models in finance domain. Question-answering and sentiment analysis tasks are two of the most widely used tasks for evaluation of language models in finance domain. Therefore, these two tasks will be used in this study to evaluate the open-source LLMs and present a comparative analysis. Timeframe allocated for this study and compute resources required to fine-tune the multiple LLMs have also been considered when selecting the tasks for evaluation.

2.4.1 Sentiment Analysis in Finance Domain

(Mishev et al., 2020) presents a survey on sentiment analysis in finance domain. As per this study, sentiment analysis is a process of identifying and categorising the opinions expressed via text with a primary aim of determining if the writer’s attitude towards a product or service is positive, neutral or negative. This study states that there are publicly available sentiment-annotated datasets which are related to movies and products. There are models (Devlin et al., 2019) which use these datasets and achieve good performance. Study also mentions that application of these models in other domains is difficult because each domain has different set of words for emotion expression. Study states that sentiments expressed in news and social media have an influence on stock prices and brand reputation and hence constant tracking and measurement of these sentiments is important for stakeholders in finance domain.

As per this study, existing research has focussed utilizing sentiment analysis on financial news (Souma et al., 2019) to predict stock prices from foreign exchange and global financial market trends (Curme and Stanley, 2015). The study (Mishev et al., 2020) states financial sector uses its own jargon, and it is not advisable to perform generic sentiment analysis because many words differ in meaning when used in the context of finance. The word ‘liability’ has a negative meaning in general sense but in finance domain it has a neutral meaning. (Loughran and Mcdonald, 2011) states that word lists which are developed for other domains misclassify words in financial domain texts. Hence, authors of this study created a lexicon of expert annotated positive, neutral, and negative words for finance which give a better picture of sentiments in financial texts. (Ghiassi et al., 2013) also introduces a twitter-specific lexicon of words for finance domain sentiment analysis.

(Wang et al., 2015) states that ML methods have been used to extract insights based on sentiment analysis from datasets of tweets. This study performs sentiment analysis on stockTwits tweets datasets and show that SVM classifier outperforms decision trees and Naïve Bayes classifier. (Mishev et al., 2020) states that conventional ML approaches are unable to extract complex features and these tasks require deep-learning approaches which enables location identification, feature extraction and order information. This study mentions that deep learning methods cascade multiple layers of non-linear processing units to extract complex features and transform them. The output from previous layer is given as input to the successive layer and this enables extracting complex features. This can be useful for generating learning patterns and understanding relationships beyond immediate neighbours. This study states that main reason for success of deep-learning approaches is due to the introduction and improvement of ways in which text is represented. Word and sentence encoders convert sentences/words into vector representation which makes it suitable for input to neural network models. As per the study, these representations help retain the semantic information in words and sentences which is critical for sentiment extraction. Study also states that techniques such as transfer learning has significantly improved the performance in sentiment analysis tasks involving financial news and texts.

The literature review in this section shows the importance of sentiment analysis to finance domain and hence sentiment analysis will be one of tasks on which LLMs will be evaluated. Financial phrase bank dataset introduced by (Malo et al., 2014) will be used in the proposed study to fine-tune LLMs for sentiment analysis task. This dataset was constructed by collecting news articles related to finance and company press releases and annotators with business knowledge classified the texts as positive, negative or neutral. This dataset can be used as benchmark for evaluating models on sentiment analysis tasks.

2.4.2 Question-answering task in Finance Domain

The main aim of question-answering task is to provide correct and precise answers to user's questions in natural language (Zhu et al., 2021b). The study states that there are two types of QA i.e., textual QA and knowledge base QA. As per the study, textual QA involves retrieving answers from unstructured text documents and knowledge base QA involves generating answers from structured documents. The study states that textual QA is analysed based on two settings i.e., Open-domain QA (OpenQA) and Machine Reading Comprehension (MRC). In MRC, context and a question are given, and QA model has to generate the answers. Context is not provided in OpenQA tasks, the model must find out the context based on the question and use the context to generate the answer. This study mentions that traditional OpenQA systems follow three stages – question analysis, document retrieval and answer extraction. Question analysis reformulates the question to generate search queries to identify the relevant documents in document retrieval stage. In document retrieval stage, the system uses the search queries from question analysis stage to extract the relevant documents. The final answer is extracted from relevant documents in answer extraction stage.

(Li et al., 2022a) proposes a framework known as Finmath which uses a tree-structured neural model to perform multi-step numerical reasoning process. This study states that financial reports typically contain both text and tabular data and models require complex numerical reasoning abilities to extract required information to perform quantitative analysis. TAT-QA (Tabular And Textual Question Answering) is a question-answering dataset that has questions that require numerical reasoning over financial reports containing both text and tabular data (Zhu et al., 2021a). This study states that state-of-the-art models on TAT-QA dataset can perform well on pre-defined aggregation operators like division and multiplication but might fail to answer queries which require multi-step numerical reasoning. This study states that Finmath framework can understand the hybrid context of financial reports and gather supporting evidence and tree-structured neural model is used to perform multi-step numerical reasoning.

(Chen et al., 2021a) introduces a dataset known as FinQA (Question Answer pairs over Financial reports). This study mentions that financial question-answering is difficult than general question-answering because it involves extracting information from heterogenous sources like tables and unstructured text. The FinQA dataset contains 8281 financial question-answer pairs along with their numerical reasoning processes. Eleven finance experts constructed this dataset from earnings reports of S&P 500 companies. The authors mention that FinQA is the first dataset of its kind to have complicated question-answer pairs based on real-word financial documents.

The literature review in this section shows the importance of question-answering task for finance domain. Therefore, question-answering task will be one of tasks on which LLMs will be evaluated in this study. FinQA dataset will be used in this study to evaluate the open-source LLMs on question-answering task.

2.4.3 Evaluation of LLMs

Evaluation of LLMs is the final and crucial part of this study as it enables a comparative analysis of different LLMs. The evaluation metrics which were used in existing literature for question-answering (Wu et al., 2023a) and sentiment analysis tasks (Arbane et al., 2023) will be used for evaluation in this study. Exact match accuracy is used to evaluate question and answering task as numerical reasoning capability of models is being evaluated as part of this task. Therefore, value predicted by the model is compared against actual value and accuracy is calculated.

The objective of sentiment analysis task is to predict the sentiment of a given text (positive, negative, or neutral). Precision measures the proportion of predicted positive instances that are actually correct, and recall measures the proportion of actual positives that have been accurately predicted by the model. F1 score considers both precision and recall and it is robust to class imbalance and gives an accurate picture of model's performance and hence F1 score will be used as evaluation metric for sentiment analysis task.

2.5 Research gap

Literature review reveals that LLMs have been the latest and most exciting development in field of NLP. The introduction of transformers architecture has been a major turning point in the evolution of neural network models as it gave rise to development of LLMs. Further review indicates that commercial LLMs like GPT series of models released by OpenAI have shown state-of-the-art performance on variety of NLP tasks. Commercial LLMs require the user queries to be sent to their server for providing the prediction. Companies and enterprises are very particular about data security and privacy and hence do not want to send data outside their own servers. This makes it difficult for enterprises to adopt commercial LLMs. This issue can be mitigated by open-source LLMs which makes it possible for enterprises to host the LLMs in their own server infrastructure. Moreover, there is no cost involved in using open-source LLMs and hence it would help companies with limited budget.

Literature review also highlights finance domain can be benefit in multiple ways using LLMs. Data security is one of the most important factors considered by stakeholders in finance domain before adopting any new technology since finance domain deals with sensitive financial information about people and businesses (Alghazo et al., 2017). Literature review indicates that there are commercial LLMs like BloombergGPT which have shown exception performance on finance domain tasks, however, require user data to be sent to their server for processing. This makes it difficult for enterprises in finance domain to adopt BloombergGPT. Therefore, open-source LLMs provide an effective alternative as it ensures data security for finance domain enterprises.

Literature review reveals that there are many open-source LLMs available and fine-tuning them leads to better results for tasks related to specific domain like finance or medicine. Comparative analysis is required to decide which LLM can be used for specific domain. The review of existing studies on LLMs on finance domain show that they are related to fine-tuning a particular LLM on finance data. To the best of author's knowledge, there are no other studies which focus on fine-tuning multiple open-source LLMs on finance data and perform comparative analysis. Therefore, this study aims to bridge that gap by fine-tuning three popular open-source LLMs i.e., LLaMA-2 7B, Mistral 7B and Falcon 7B on finance domain data. Followed by evaluating them on key NLP tasks (question-answering and sentiment analysis) to present a comparative analysis. The stakeholders in finance domain can use the results of this study of choose appropriate open-source LLM for their use-case.

2.6 Summary

Literature review for this study begins by giving an overview of NLP pre-processing and feature extraction techniques (section 2.2). Neural network models leading up to the development of LLMs are also discussed in this section. Studies on evolution of LLMs are discussed in section 2.3. Overview on GPT series of models and major drawbacks of commercial LLMs are discussed in this section. Studies which introduce the open-source LLMs are summarised in this section. This study is focussed on finance domain and hence literature on language models in finance domain is discussed in this section. The need for fine-tuning LLMs on domain specific data and instruction tuning techniques are also discussed in this section. The key NLP tasks which will be used for fine-tuning and evaluating the LLMs are discussed in section 2.4. Literature review on evaluation metrics is also discussed in this section. Section 2.5 provides overview of research gaps identified and contributions of this study. Section 2.6 summarises the literature review.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

The main objective of this research is to fine-tune open-source foundation LLMs on financial domain data and present comparative analysis. Hence, research methodology chapter of this study gives the details about the various data preparation steps, LLMs, research methods, fine-tuning and evaluation techniques used in this research. The relevance of research methods (discussed in this chapter) to the overall objectives of the study is also mentioned. Fig. 3.1 gives a visualisation of steps involved in this research. Various sections in this chapter are aligned with this research workflow.

Section 3.2 focusses on explaining the details about data preparation. Details about collection of datasets used in this study is provided. The importance and relevance of these datasets to the NLP tasks considered in this study are also discussed in this section. The steps used to convert the data into a format which can be used by model for training are discussed in this section. The conversion of the training dataset into instruction tuning dataset along with importance of instruction tuning is also discussed in this section.

Transformers architecture is the basic building block of LLMs and hence section 3.3 delves into the details of transformers, techniques used to develop it and reasons for LLMs adopting transformers. Open-source foundation LLMs are an important part of this study and hence three LLMs considered for this research are LLaMA-2 7B, Mistral 7B and Falcon 7B. These are are explained in detail in section 3.4. The datasets used to train these LLMs and unique features are also discussed. The criteria for selecting these three LLMs from multiple open-source LLMs available is also discussed.

Fine-tuning is the most important part of this study and hence various fine-tuning techniques used in this study are discussed in detail in section 3.5. LoRA fine-tuning technique has been discussed in detail in this section since it is be used to fine-tune LLMs in this work. The methodology behind LoRA technique along with its importance are discussed. This section also discusses about QLoRA technique which is also used along with LoRA to effectively fine-tune LLMs. The evaluation metrics used in this study to compare the LLMs are discussed in section 3.6.

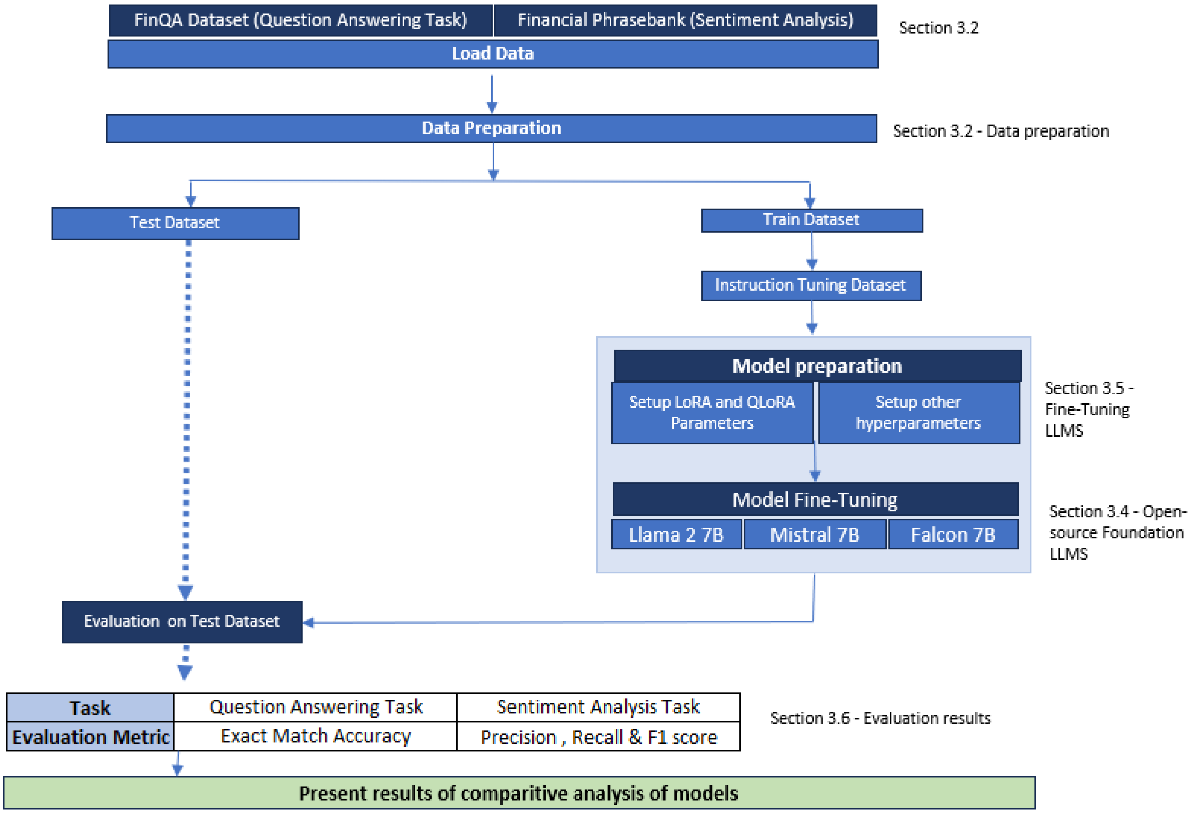

FinQA Dataset (Question Answering Task)
Financial Phrasebank (Sentiment Analysis)
Load Data
Section 3.2
Data Preparation
Section 3.2 - Data preparation
Test Dataset
Train Dataset
Instruction Tuning Dataset
Model preparation
Setup LoRA and QLoRA Parameters
Setup other hyperparameters
Section 3.5 - Fine-Tuning LLMS
Model Fine-Tuning
Llama 2 7B
Mistral 7B
Falcon 7B
Section 3.4 - Open-source Foundation LLMS
Evaluation on Test Dataset
Task
Question Answering Task
Sentiment Analysis Task
Evaluation Metric
Exact Match Accuracy
Precision , Recall & F1 score
Section 3.6 - Evaluation results
Present results of comparitive analysis of models

Figure 3.1 Workflow of steps involved in the study

Methodology used in this research involves following steps (see Fig. 3.1):

- Loading the required datasets (FinQA and Financial Phrasebank) for respective tasks (question answering and sentiment analysis).
- Converting the data into a format which can be used to train (fine-tune) the LLMs.
- Constructing instruction tuned datasets by overlaying the prompts (instructions) on train dataset.
- Building the framework required to fine-tune the model.
- Selecting the relevant hyperparameters for fine-tuning through hyperparameter tuning.
- Fine-tuning the selected open-source LLMs on train dataset.
- Evaluating the models using relevant metrics for respective tasks to provide a comprehensive comparative analysis of LLMs for finance domain.

3.2 Data Preparation

This research is focussed on fine-tuning open-source LLMs on finance domain data and evaluating them on question-answering and sentiment analysis tasks. Hence, financial domain datasets for these tasks have been selected and used in this study. Research involves multiple NLP tasks. There is no single dataset which can be used for different tasks and hence comparative studies of LLMs use different dataset for each task (Wang et al., 2023). Therefore, separate datasets for question-answering and sentiment analysis tasks are used in this work. This section gives an overview of literature on construction of these datasets and rationale for selecting these datasets in this research. FinQA dataset is used for question-answering task and Financial phrasebank dataset is used for sentiment analysis task.

(Chen et al., 2021a) has released FinQA dataset as part of their study to evaluate financial numerical reasoning abilities of pre-trained models. FinQA dataset consists of 8,821 question answer pairs created based on financial reports, dataset contains both text and tabular data and questions are framed in a manner which requires numerical reasoning to arrive at the answers. This dataset is a benchmark dataset and has been used by many studies (Mavi et al., 2023; Zhao et al., 2023b; Zhu et al., 2023). Therefore, this dataset a good to be used in this study to fine-tune LLMs and evaluate them on question-answering task. Logical and numerical reasoning capabilities are important from finance domain perspective apart from general question-answering capabilities and this dataset will test these abilities of LLMs considered in this study.

(Malo et al., 2014) has released Financial phrasebank dataset as part of their study which explores how semantic orientations for sentiment analysis can be better detected in financial and economic news. Financial phrasebank dataset consists of 2259 sentences from financial press releases and news which were tagged as positive, negative, and neutral sentiments. Similar to FinQA dataset, Financial phrasebank dataset is also a benchmark dataset and widely used in many studies related to sentiment analysis tasks (Yang et al., 2020b; Leippold, 2023; Shang et al., 2023). Therefore, this dataset is used in this study to fine-tune LLMs and evaluate them on sentiment analysis task. Details about dataset description and dataset links are provided in the Chapter 4 - Analysis and Design.

The datasets used in this study do not require additional data cleansing and pre-processing steps (like imputation) since these are already cleansed and pre-processed as part of the study which introduced these datasets (Malo et al., 2014; Chen et al., 2021a). However, data in these datasets needs to be converted into a format which can be used for fine-tuning and evaluating the LLMs. Inputs to LLMs are given using constructs known as prompts (Zamfirescu-Pereira et al., 2023). For example, in question-answering tasks, the prompt used to fine-tune the LLM will contain the context (paragraph on which LLMs will be questioned), question based on context provided and the actual answer. Similarly for sentiment analysis task, prompt used for fine-tuning will contain the context (the input paragraph for which sentiment needs to be determined), question and actual sentiment of input text. LLMs use this to understand the relationship between the input and output so that it can produce accurate results during inference. There are two methods of prompting the LLMs during evaluation – zero-shot and few-shot. Zero-shot prompting (Zhong et al., 2021) involves giving the context (input text) and corresponding question in the prompt and LLM generates inferences (answers) based on the same. Few-shot prompting (Schick et al., 2022) like zero-shot prompting has context and question and in addition to this it has few examples of input text, question and actual answers. This study will use zero-shot prompting technique when evaluating the LLMs since the models are already fine-tuned on question-and-answer pairs.

Hugging Face repository which hosts the datasets has already split the datasets into train and test instances and hence no additional split will be done as part of this study. The next step in data preparation involves instruction-tuning (Wei et al., 2021) on train dataset and converting them into instruction-tuning dataset. Instruction-tuning involves adding natural language instructions (describing the task to be performed) to the prompt template which will be used for fine-tuning the LLMs. The study which introduces instruction-tuning technique reports that it improves zero-shot performance of LLMs significantly. Hence, instructions relevant to the tasks considered for this research will be added to prompt used for fine-tuning. The modified prompt will have instruction of the task followed by the context, question, and actual answer. The dataset which consists of these prompts is known as instruction-tuning dataset and will be used to fine-tune the LLMs in this research. Instruction-tuning will be done only on train dataset since that is being used for fine-tuning the LLMs. Examples of prompts and details about exact instructions added to the prompts (based on the task) will be given in Chapter 4 - Analysis and Design.

3.3 Transformers

The introduction of transformer neural network architecture (Vaswani et al., 2017) proved to be a breakthrough in the development of language models. This study states that key advantage of transformers over other neural network architectures like RNN is that it enables parallelism during training. As per the study, neural networks like RNN process the data sequentially due to recurrent components in the architecture. The transformers architecture allows for parallelism, and this enables training the language models on massive amounts of text data which is essential for development of LLMs. This is the main reason LLMs adopt transformers model as foundational architecture over other neural network models. This section gives an overview of architecture of transformers and techniques that enable parallelism. Fig. 3.2 gives the overall architecture of transformers.

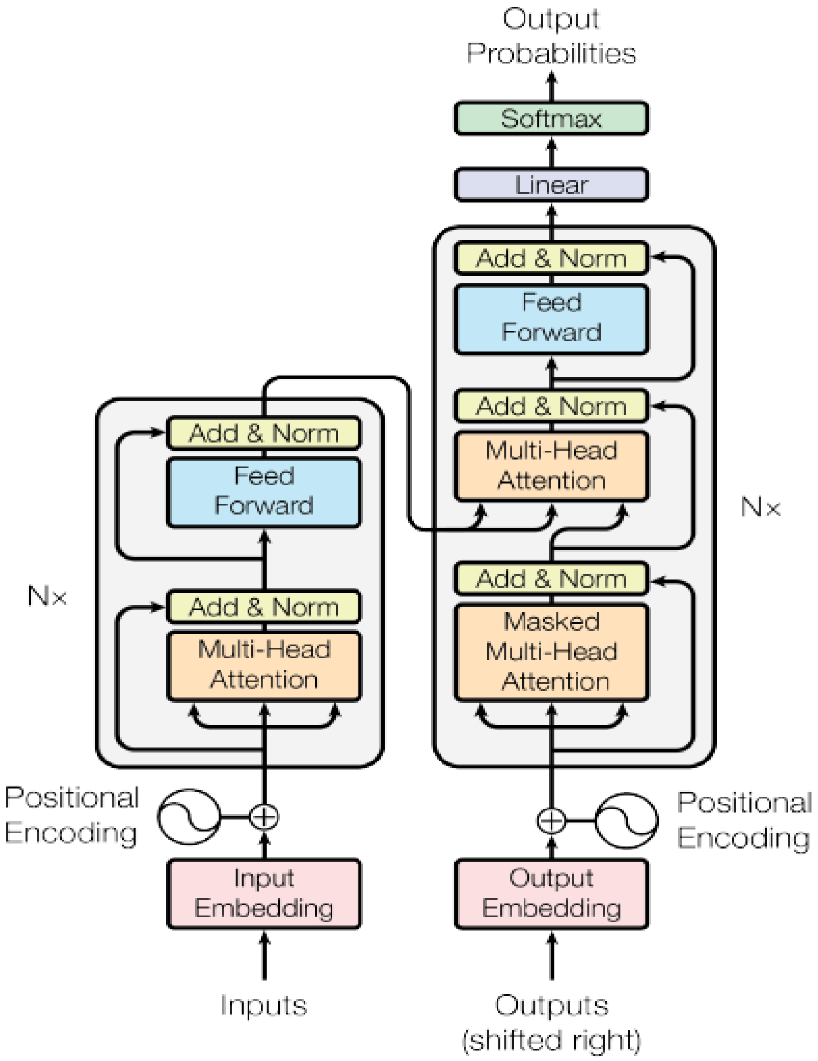


Figure 3.2 Transformers Architecture (Vaswani et al., 2017)

As per the study which introduces transformers, there are two main components – encoder and decoder. The study states that first step in the encoder unit involves converting input text to word embedding vectors and adding positional encoding values. Positional encoding has been used by transformers to keep track of order of words in a sequence. The authors of the study state that positional encoding is required since transformers does not contain any recurrent components, some information must be injected about the relative and absolute position of words in a sequence. Positional encoding values are generated using sine and cosine functions of different frequencies and these values have the same dimension as the input word embeddings. Positional encoding is followed by self-attention layer in the encoder which implements the attention mechanism. Attention mechanism captures the relationship between a particular word and all other words in the sequence. Fig. 3.3 gives an overview of attention mechanism.

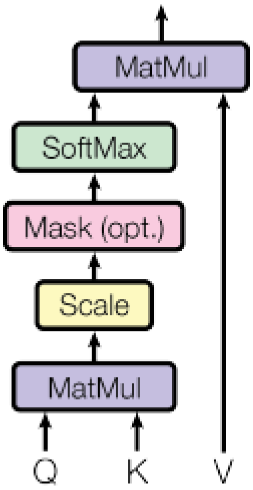


Figure 3.3 Self-Attention (Vaswani et al., 2017)

The self-attention layer comprises of three main components - query vectors(Q), key vectors (K) and value vectors (V). The study states position encoded vectors obtained from first step is multiplied by three separate weight vectors to obtain the query, key, and value vectors respectively. The query vector of a particular word in the sequence has been multiplied by key vectors of all other words in the sequence (see MatMul in Fig. 3.3). Output vector obtained from this step has been scaled and softmax function applied which results in a vector with a set of values for a particular input word. This output vector represents the similarity of a particular word with all other words in the sequence. This output vector was multiplied by value vector to obtain self-attention values for a particular word.

These values were named as self-attention values by authors of the study since they capture information about relationship of particular words (self) with all other words in the sequence. The weight vectors used to calculate query, key and value vectors for a particular word can be re-used for all other words in the sequence as per the study. This feature of the architecture enabled calculating self-attention values of all words in a sequence in a parallel manner instead of calculating it in sequential manner one-by-one. This allowed the transformers model to be trained on large volumes of text since it can take advantage of parallel computing for faster processing. This unit with set of weight, query, key, and value vectors was named as self-attention layer in the study.

The stack of multiple self-attention layers together known as multi-head attention layer (as shown in Fig. 3.1) to capture relationship between words for complicated sentences and paragraphs. Self-attention values were added to position encoded vectors to get residual values and these values were passed through a feed forward neural network to obtain the output of encoder unit. As per the study, these output values of encoder unit can capture both the positional information of words in a sequence and encode the information about relationship between words. The output values of encoder unit were given as input to second multi-head attention layer in the decoder unit (as shown in Fig. 3.1)

The decoder unit is like that of encoder unit with the only difference being, two multi-head attention layers were used in the decoder unit (as shown in Fig. 3.1). First multi-head attention layer is like that of encoder unit and is used to capture the relationship between words in the output sequence. The output of first multi-head attention layer has been given as input to second masked multi-head attention layer. The second masked multi-head attention layer has been used to capture the relationship between input and output sequences. This is achieved by implementing attention mechanism on encoded input sequences (obtained from encoder unit) and output sequences obtained from first multi-head attention layer. The output of second-multi head attention layer captures two things - relationship among words in the output sequence and relationship between input and output sequences. As per Fig. 3.1, this output has been passed to feed forward network and softmax function which results in producing output probabilities. The authors report that transformers models were trained significantly faster than models based on recurrent or convolutional techniques. They also mention that transformers model was evaluated on translation tasks and was able to outperform the state-of-the-art models. The open-source foundation LLMs used in this study i.e., LLaMA-2, Mistral and Falcon are built based on the transformers architecture.

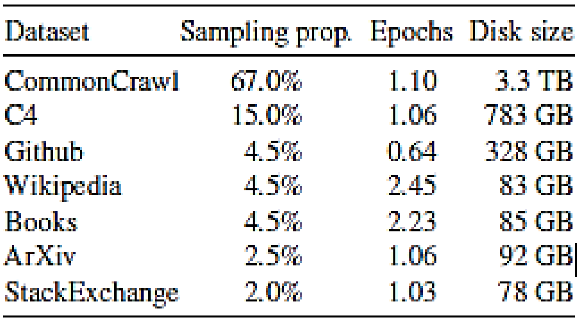
3.4 Open-Source Foundation LLMs

Foundation LLMs are language models trained on vast amounts of data and capable of performing wide variety of NLP tasks (Zhou et al., 2023). The main objective of this research is to fine-tune the open-source foundation LLMs on finance data and perform comparative analysis and hence section focusses on detailed overview of LLMs used as part of this study.

3.4.1 LLaMA-2:

LLaMA-1 released by Meta is a collection of foundation language models with parameter sizes ranging from 7B to 65B (Touvron et al., 2023a). This model shows that state-of-the-art LLMs can be developed by using publicly available datasets for training instead of using private or proprietary datasets. Data from variety of public sources have been used for training. Table 3.1 shows different publicly available data sources used for training. It also shows the proportion of each dataset used in training along with the size and number of epochs during training.

Table 3.1 – Data source used for Training LLaMA-1 (Touvron et al., 2023a)



Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Common crawl is an open repository of web pages consisting of petabytes of data collected since 2008. Common crawl dumps ranging from 2017 to 2020 have been used to train LLaMA-1. Authors of LLaMA-1 model use CCNet pipeline (Wenzek et al., 2019) to de-duplicate data at line level, remove non-English pages and filter low quality content. In addition to this, a linear model was used to categorize pages used as references in Wikipedia vs random web pages and pages which were not used as references were discarded. The authors state that during their exploratory experiments, it was observed that using diverse common crawl datasets improved performance and hence C4, a cleansed version of common crawl dataset has also been included in the training dataset. The public GitHub dataset was also used for training. The projects which were kept under public licenses like Apache were only considered by the authors for the study. Wikipedia dumps from June to August 2022 were considered and pre-processing was done to remove comments and hyperlinks. Gutenberg dataset which consisting of books in public domain has also been included in the dataset. Authors included data from Arxiv website to make sure scientific data is also used as part of training process. Authors also included StackExchange data which consists of question and answers across various domains ranging from chemistry to computer science. Byte pair encoding (BPE) algorithm (Zouhar et al., 2023) has been used to tokenise the training data. Study states that dataset after tokenisation consists of 1.4 trillion tokens.

Authors state that transformers architecture (explained in detail in section 3.4) has been used as basic building block of LLaMA-1 model and pre-normalisation (Zhang and Sennrich, 2019), SwiGLU activation function (Shazeer, 2020) and rotary embedding techniques (Su et al., 2021) were added to improve upon the architecture.

LLaMA-1 model has been evaluated on various NLP tasks as part of this study. Eight standard common sense reasoning benchmark datasets were used for evaluation. Performance of LLaMA-1 was measured against three models – PaLM, Chinchilla and GPT-3. Study reports that LLaMA -1 (65B) outperforms Chinchilla on all benchmark datasets except one. Study also reports that LLaMA 1 (13B) was able to outperform GPT-3 on most benchmarks. LLaMA-1 model was evaluated on two closed-book question answering datasets and study reports that LLaMA-1 (65B) model was able to achieve state-of-the-art performance. LLaMA-1 was also evaluated on code generation tasks and study reports that it was able to outperform other general models like PaLM and LaMDA.

LLaMA-2 (Touvron et al., 2023b) released by Meta is an updated version of LLaMA-1 model with parameter sizes ranging from 7B to 70B. As per the study, training data for LLaMA-2 comprises of datasets which were used to train LLaMA-1 model and in addition to this, a new mix of data from public sources was added. The study also clarifies that this training data does not include data from Meta's products or services. Study also mentions that data obtained from certain websites that were known to contain high volume of personal information were removed from training dataset. Study states that training dataset for LLaMA-2 model contains two trillion tokens compared to LLaMA-1 model which comprises of 1.4 trillion tokens. The tokenizer (BPE) used for LLaMA-1 model has been re-used for LLaMA-2 model as well. Another major improvement over LLaMA-1 model is that LLaMA-2 model has context length of 4096 tokens whereas LLaMA-1 model has context length of 2048 tokens. This enables LLaMA-2 to process more information, which in turn helps with supporting longer histories for chat-based applications, summarisation tasks, and understanding longer documents. Study states that architecture of LLaMA-2 model is similar to LLaMA-1 model. However, LLaMA-2 uses grouped query attention (Ainslie et al., 2023) to improve inference scalability. This study also releases LLaMA-2 chat model which is a result of fine-tuning LLaMA-2 model using instruction tuning and RLHF techniques.

LLaMA-2 model has been evaluated on multiple NLP tasks and compared against both open-source LLMs like Falcon, MPT and LLaMA-1 and closed-source LLMs like GPT 3.5 and PaLM. NLP tasks considered for evaluation were reading comprehension, mathematical reasoning, commonsense-reasoning, and code-generation. The study reports that LLaMA-2 model was able to outperform LLaMA-1 model and it is also mentioned that LLaMA-2 7B and 30B model outperform MPT models on all NLP tasks except code generation. Authors also report LLaMA-2 7B and 34B models outperform Falcon 7B and 40B models on all benchmarks. It is also stated that LLaMA-2 70B model was able to outperform all open-source models considered for evaluation on all the benchmarks. From an evaluation perspective, LLaMA-2 model shows that it is among the best open-source LLMs. Hence, LLaMA-2 is one of the open-source foundation models considered for this study which involves fine-tuning and comparing open-source LLMs for finance domain.

3.4.2 Mistral:

Mistral LLM (Jiang et al., 2023) released by Mistral AI is a 7B parameter open-source foundation model engineered for efficiency and superior performance. The study which introduces Mistral 7B model states that LLMs with large number of parameters are required to achieve higher performance. However, LLMs of large parameter sizes requires lots of computational resources and could also exhibit inference latency. The study states that this could act as a barrier for deployment in real-word scenarios. The study states that it is critical to develop balanced models with a relatively lesser number of parameters which can deliver high-level of performance and efficiency. As per the study, Mistral uses transformers as basic building block of the model architecture, however, adds features like Grouped Query Attention- GQA(Ainslie et al., 2023) and sliding window attention (SWA) (Beltagy et al., 2020). The study states that GQA is used to accelerate inference speed and reduce memory requirement during decoding, thus allowing for higher throughput which is an important factor in real-time applications. Study also states that SWA can help with handling longer sequences effectively at reduced computational cost. The study mentions GQA and SWA attention mechanisms lead to enhanced efficiency and performance of Mistral model. The information about public datasets which were used for pre-training the model has not been revealed by authors of the study which introduces the Mistral model. Another contribution of this study is development of Mistral 7B Instruct model. The study mentions that Mistral 7B Instruct model was developed by fine-tuning Mistral 7B model on publicly available instruction tuning datasets available on hugging face repository.

As per the study, evaluation of Mistral 7B model was done using multiple NLP tasks like commonsense reasoning, reading comprehension, numerical reasoning, code generation and benchmark datasets which consists of data for multiple tasks. Mistral model is compared against LLaMA-2 7B, LLaMA-2 13B and Code-LLaMA 7B (Rozière et al., 2023) models. Code-LLaMA is a collection of models released by meta which takes LLaMA-2 model as the base and fine-tunes it on code. The study reports that Mistral 7B model outperforms LLaMA-2 7B and LLaMA-2 13B models on all metrics. The study also reports that Mistral was able to match the performance of code-LLaMA 7B model on code generation task without sacrificing performance on other non-code related tasks.

The evaluation results reported by the study that introduces Mistral 7B shows that it is among the best performing open-source foundation models, hence Mistral 7B is one of the models which will be used as part of this research. This research only considers foundation models trained from scratch and hence Mistral 7B instruct model which is a fine-tuned model is not considered for this research.

3.4.3 Falcon:

Falcon developed by TII is a collection of open-source foundation LLM with parameter sizes ranging from 7B to 180B (Penedo et al., 2023). The study which introduces Falcon LLM states that unique feature of Falcon LLM is that it has been trained on curated web data alone.

The scaling laws (Aghajanyan et al., 2023) state that increase in model size and size of dataset used for training (known as scaling) results in improvement in performance of LLMs (known as emergent abilities). The study states that large datasets used to train LLMs are a mix of web data and data from other sources like books, technical documents, and social-media conversations. As per the study, curation of these large datasets plays an important role in producing performant models, however, curation is a labour-intensive process requiring specialised processing yielding limited amount of data. Study also states that high proportion of pre-training dataset used to train LLMs are sourced from massive web-crawls which can be used with limited human intervention. However, the quality of this data has been inferior to that of manually curated sources. To mitigate this issue, study which introduces Falcon LLM uses efficient processing techniques to curate web data with minimal human intervention and produce high quality web dataset.

The study states that processing techniques used to curate the web data are implemented through a pipeline known as MDR (MacroData Refinement). The study mentions that first step of this pipeline involves filtering relevant URL to exclude data from fraudulent websites. The next step of the pipeline involves text extraction to extract only main content of webpages ignoring headers and footers. As per the study, next step in the pipeline is to filter and remove webpages which do not have natural language text and it is followed by step to documents with excessive paragraph and line repetitions. This is followed by step to remove webpages that are machine generated spam. The study states that final step of the pipeline is deduplication, and it involves removing documents (web pages) which are similar. The high-quality web dataset produced by processing raw web data using this pipeline is called as REFINEDWEB. This dataset has been used to train the Falcon LLM.

Falcon LLM is based on transformer architecture. Falcon LLM has been evaluated on 18 benchmark datasets across various NLP tasks against GPT-3 models, GPT-Neo(X/J) models, OPT series of models, PaLM and Pythia models. The study states that the models used for comparison against Falcon LLM have been trained on manually curated corpora. Falcon LLM was able to outperform other LLMs considered for evaluation.

The evaluation results reported by this study show that Falcon LLM is among the best open-source foundation models and hence it will be one of the models used in this research. There are multiple Falcon models with parameter sizes ranging from 7B to 180B. Falcon 7B model will be used in this research considering the resources required to train the model (GPU compute units) and timeframe allocated for this study.

3.5 Fine-Tuning LLMs

Supervised fine-tuning is a method of transfer learning where a base model which is good at a particular task is trained on another related task (Zhuang et al., 2019). Research (Gururangan et al., 2020) shows that language models fine-tuned on data for a specific domain performs better on that tasks on that domain compared to language models which have not been fine-tuned. (Ding et al., 2023) gives an overview of fine-tuning techniques and this study mentions that full parameter fine-tuning involves initializing the model with pre-trained weights and updating all the parameters and producing separate instances for different downstream tasks. Study also states that LLMs have billions of parameters and hence full parameter fine-tuning would involve updating all these parameters. This is difficult to implement because of cost of deployment and computational resources required to perform full parameter fine-tuning. To mitigate this issue, LLMs can be fine-tuned by optimization of few parameters instead of fine-tuning billions of parameters and this gave rise parameter-efficient fine-tuning techniques. Adapter tuning (Houlsby et al., 2019b) was among the initial approaches to fine-tune LLMs using limited number of parameters. This technique involves inserting adapter modules with bottleneck architecture between layers in LLMs and only parameters for these modules would be fine-tuned. However, the major disadvantage with adapter tuning was that it introduced inference latency since the adapter layers need to be processed sequentially. LoRA was introduced to overcome this limitation (Hu et al., 2021).

3.5.1 LoRA

Neural networks generally have dense layers and weights associated with these layers are stored in form of matrices. The main idea behind this technique is that these weight matrices can be approximated well by matrices of low rank without losing crucial information. Rank of a matrix is the number of linearly independent rows or columns in the matrix. For example, let’s consider matrix M as given below. This is a matrix with 5 rows and 5 columns; however, the rank of this matrix is 2 because the first and fourth column are identical and dependent on each other and similarly fifth column can be derived from third column.

$$M = \begin{bmatrix} 19 & 9 & 12 & 19 & 8 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 3 & 0 \\ 6 & 2 & 0 & 6 & 0 \\ 25 & 11 & 12 & 25 & 8 \end{bmatrix}$$

LoRA study uses rank decomposition to represent the original weight matrix as product of two matrices with lower dimensions. The matrix M(5x5) can be represented as dot product of following matrices S(5x2) and P(2x5) using rank decomposition.

$$M = \begin{bmatrix} 19 & 9 & 12 & 19 & 8 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 3 & 0 \\ 6 & 2 & 0 & 6 & 0 \\ 25 & 11 & 12 & 25 & 8 \end{bmatrix}$$

$$S = \begin{bmatrix} 4 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 2 \\ 4 & 3 \end{bmatrix}$$

$$P = \begin{bmatrix} 4 & 2 & 3 & 4 & 2 \\ 3 & 1 & 0 & 3 & 0 \end{bmatrix}$$

The total number of values in decomposed matrices S and P is 20 which is lesser than the number of values (25) present in the original matrix. The rank decomposition done in this example was based on the original rank of the matrix. LoRA states that even a rank which is lower than rank of the original matrix can be used for decomposition of the original matrix and still retain the information of the original matrix. Choosing a lower rank led to even lesser number of values in the decomposed matrices and this is particularly effective in case of LLMs which comprises of billions of parameters. For example, considering a weight matrix C in LLM with dimension of 1000 rows and 1000 columns. This matrix will have 1 million parameters. If rank of 8 is considered in this example, the number of parameters in the decomposed matrices (I and J) will have a total of 16,000 parameters.

C(1000x1000) = 1 million parameters

Performing rank decomposition using rank =8

C(1000x1000) = I(1000x8) J(8x1000) = 8000+8000 = 16,000 parameters.

This example shows the impact of LoRA to reduce the number of trainable parameters. The LoRA study implemented this technique on GPT-3 model which has 175 billion parameters and reported that it resulted in reduction of number of trainable parameters by 10,000 times and GPU memory requirement by 3 times.

LoRA implementation approach is shown in Fig. 3.4. The original pre-trained weights W of the base model are frozen, and no updates are made. A and B represent the decomposed low rank matrices which were obtained based on chosen rank r. A and B matrices contain the trainable parameters. Once fine-tuning process is completed, the A and B matrices are added back to the original weight matrix W.

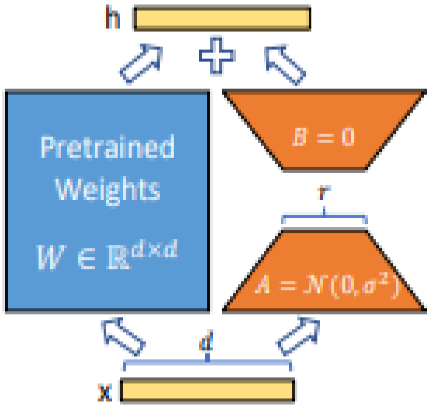

h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 3.4 LoRA implementation (Hu et al., 2021)

RoBERTa, BERTa, GPT-2, and GPT-3 models were fine-tuned using LoRA techniques and comparison is performed against fully fine-tuned version of these models. It is reported that LoRA fine-tuned models perform on-par or better than fully fine-tuned models despite having fewer number of trainable parameters. This shows the power of LoRA techniques to reduce the number of trainable parameters and hence it is crucial part of this research and will be used to fine-tune the LLMs in this study. Rank(r) is an important hyperparameter for LoRA and hence exploration will be part of this research to identify the optimal rank for LoRA configuration.

QLoRA (Dettmers et al., 2023) is another PEFT technique which can be used in conjunction with LoRA to reduce the number of computational resources required to fine-tune LLMs. QLoRA uses quantization techniques to save memory without compromising on performance.

The study defines quantization as a process of discretizing an input which holds more information to a representation with less information. This means taking a data type with more bits and converting into fewer bits. The weight matrices used in the transformers architecture in LLMs are stored in the form of 16-bit floating point numbers. QLoRA uses quantization techniques to convert these 16-bit numbers into quantized 4-bit numbers without losing crucial information. This is achieved by using 4-bit NormalFloat (NF4) quantization. This study considers LLaMA, RoBERTa, Alpaca, OPT and Pythia models and QLoRA fine-tuned versions and fully fine-tuned versions of these models are compared across various NLP tasks. Study reports that performance of QLoRA fine-tuned models is on par with fully fine-tuned models. The study states that QLoRA reduces the average memory requirement of fine-tuning a 65B LLM from 780GB to lesser than 48GB without degrading the predictive performance when compared with fully fine-tuned baseline. This shows the importance of QLoRA in the context of fine-tuning and hence will be used in this research to fine-tune the LLMs.

3.6 Evaluation of LLMs

Evaluation is the final part of this research and results of evaluation are used to present a comparative analysis of open-source foundation LLMs for finance domain. The evaluation metrics selected for question-answering and sentiment analysis tasks were based on existing literature (Arbane et al., 2023; Wu et al., 2023a) on these tasks.

The question-answering task involves evaluating the numerical reasoning capabilities of LLMs. The questions in this task are framed in such a way that LLMs need to extract the required numbers from given input text and perform numerical operations to arrive at the answer. For example, text extracted from annual report of a company with net profits for different years will be given as input to the LLM and question will be ‘what is % increase in net profit in 2023 compared to previous year?’. The actual answer is typically an exact number and hence exact accuracy metric is used to evaluate the LLMs on this task. Exact match (EM) accuracy for a record in a dataset will be 1 if prediction exactly matches actual answer else it is 0. This value is calculated for all records and divided by total number of records to get final EM accuracy score.

EM accuracy = Records where predictions exactly match answers/Total number of records in the dataset.

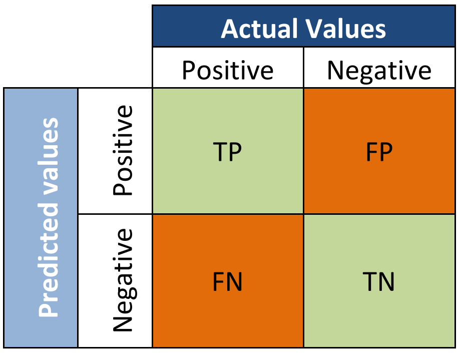
Sentiment analysis task involves predicting sentiment of a given financial text. Dataset contains three sentiment labels – positive, negative and neutral and hence this is a multi-class classification problem. Precision, recall and F1-score metrics will be used in this study to evaluate the LLMs on sentiment analysis task. Confusion matrix which is used to calculate the true positives, true negatives, false positives, and false negatives form basis of calculation of precision and recall. Table 3.2 shows the confusion matrix for a binary classification problem with two classes – positive and negative.

True Positive (TP) = Actual value is positive and predicted value is also positive

True Negative (TN) = Actual value is negative and predicted value is also negative

False Positive (FP) = Actual value is negative but predicted value is positive

False Negative (FN) = Actual value is positive but predicted value is negative



		Actual Values	
		Positive	Negative
Predicted values	Positive	TP	FP
	Negative	FN	TN

Table 3.2 - Confusion Matrix for Binary classification

$$\text{Precision} = TP/(TP+FP)$$

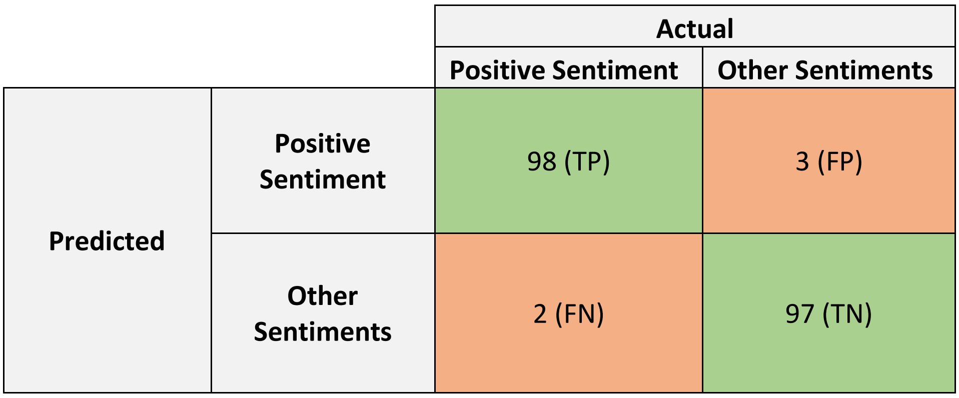
$$\text{Recall} = TP/(TP+FN)$$

Precision measures the proportion of predicted positive instances that are actually correct (TP) out of the total number of instances predicted as positive (TP+FP), and recall measures the proportion of actual positives that have been accurately predicted (TP) out of the total actual positives (TP+FN) by the model. Higher precision score indicates a low number of false positives and higher recall indicates low number of false negatives. If the aim is to reduce false negatives, the model needs to have high recall. There are instances where both false positives and false negatives are both equally undesirable and goal is to have as few false positives and false negatives as possible. However, there is a trade-off between false positives and false negatives and decreasing one increases the other. F1-score can mitigate this issue by considering both precision and recall. F1-score gives a more robust and balanced view of predictive ability of the model since it considers both precision and recall and it is particularly useful for imbalanced datasets. F1-score is harmonic mean of precision and recall.

F1-score = 2*(Precision*Recall)/(Precision +Recall)

The metrics discussed so far have been with respect to binary classification. This can also be extended to multi-class classification. There are three classes for sentiment classification tasks – positive, negative, and neutral. Table 3.3 shows confusion matrix for positive sentiment class with some sample data.

Table 3.3 – Confusion matrix for positive sentiment



		Actual	
		Positive Sentiment	Other Sentiments
Predicted	Positive Sentiment	98 (TP)	3 (FP)
	Other Sentiments	2 (FN)	97 (TN)

This confusion matrix uses one vs rest methodology where positive sentiment is considered as one class and other sentiments (neutral and negative) are considered as other class. This enables use of binary classification confusion matrix to calculate the true positives, true negatives, false positives, and false negatives metrics. These metrics will in-turn be used to calculate the precision, recall and F1-score for a particular class. Confusion matrix can be constructed similarly for negative and neutral sentiment classes. Table 3.4 shows the description of confusion matrix metrics for positive sentiment class.

Table 3.4 – Confusion matrix metrics for Positive sentiment class

Metric	Value	Description
True Positives	98	Actual sentiment is positive and LLM has also predicted the sentiment as positive
True Negatives	97	Instances where sentiment is not positive and LLM has correctly predicted sentiment as not positive (either negative or neutral)
False Positives	3	Instances where sentiment is actually negative or neutral, but LLM has predicted it as positive
False Negatives	2	Instances where sentiment is positive, but LLM has predicted as neutral or negative

F1-scores for each sentiment class can be combined to provide one metric for the model. Macro-F1 score metric will be used in this study for this purpose. Macro-F1 score involves calculating mean of F1-scores of each sentiment class. This is preferred over other F1-score metrics (like weighted average F1-score) because macro F1-score is a useful metric for imbalanced datasets as it treats all classes equally. The sentiment analysis dataset is imbalanced with close to 62% of records belonging to neutral sentiment and hence macro F1-score is the right metric for evaluation.

3.7 Summary

Data preparation is the first step in this study and hence first section (section 3.2) in research methodology describes the data. This section delves into details of collection of datasets and relevance and rationale for using these datasets for this research. This section also discusses about steps performed to convert the data in the datasets into a format which can be used to fine-tune the LLMs. The next section (section 3.3) discusses about transformers architecture in detail since it forms the basis for development of LLMs. The rationale for LLMs adopting transformers architecture has also been discussed in this section. Section 3.4 discusses in detail about three open-source foundation LLMs used in this research and this section also mentions the rationale for selecting these LLMs. Section 3.5 gives an in-depth view of fine-tuning methodologies used in this study and the final section of this chapter discusses about the evaluation methods and metrics which will be used to perform the comparative analysis.

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Introduction

The main objective of this research is to fine-tune open-source foundation LLMs on finance datasets and perform evaluation to provide comparative analysis and hence fine-tuning the LLMs are an important part of the work involved in this research. This chapter mainly focusses on steps leading up to fine-tuning of LLMs. Section 4.2 discusses about the details of datasets used and data preparation steps required to convert the records in the dataset into a format which can be used for fine-tuning LLMs. This section also discusses about adding instructions to train dataset to create instruction tuning dataset. Section 4.3 focusses on fine-tuning framework used in this research. This section gives details about various hyperparameters used for fine-tuning. This section also discusses about hyperparameter tuning process used to identify the best hyperparameters. Details about various software tools, packages, and libraries and the hardware configuration used for training (fine-tuning) the LLMs is also documented in this section.

4.2 Data Preparation

LLMs are evaluated on question-answering and sentiment analysis tasks in this study. FinQA dataset has been used for question-answering task and financial phrasebank dataset has been used for sentiment analysis. These datasets are benchmark datasets used in multiple studies and they are already cleansed and pre-processed by the studies which introduced these datasets and hence no additional processing is required. These datasets have been hosted in hugging face datasets repository and details are given in table 4.1. Hugging face datasets repository is a vast collection of datasets across various domains which can be used for machine learning tasks.

Table 4.1 – Dataset details

Dataset	Link
FinQA	https://huggingface.co/datasets/dreamerdeo/finqa
Financial phrasebank	https://huggingface.co/datasets/financial_phrasebank

FinQA dataset comprises of 7398 records with data extracted from financial reports. These reports not only contain text but also tabular information which captures the financial performance of a company across various time periods (years or quarters). Hence, this dataset contains separate columns which capture tabular and text information from financial reports along with question and actual answer. Table 4.2 gives column wise description of FinQA dataset.

Table 4.2 – FinQA dataset description

Column	Datatype	Description
id	string	Unique identifier for each record in the dataset
Post_text	list (of strings)	Text after tabular data is stored as a list of strings in this column
Pre_text	list (of strings)	Text before tabular data is stored as a list of strings in this column
question	string	Column containing the question which will provided as input to LLM
answer	string	Column containing the actual answer
gold_evidence	string	Mathematical reasoning used to arrive at the answer
table	list (of strings)	Tabular data from financial report

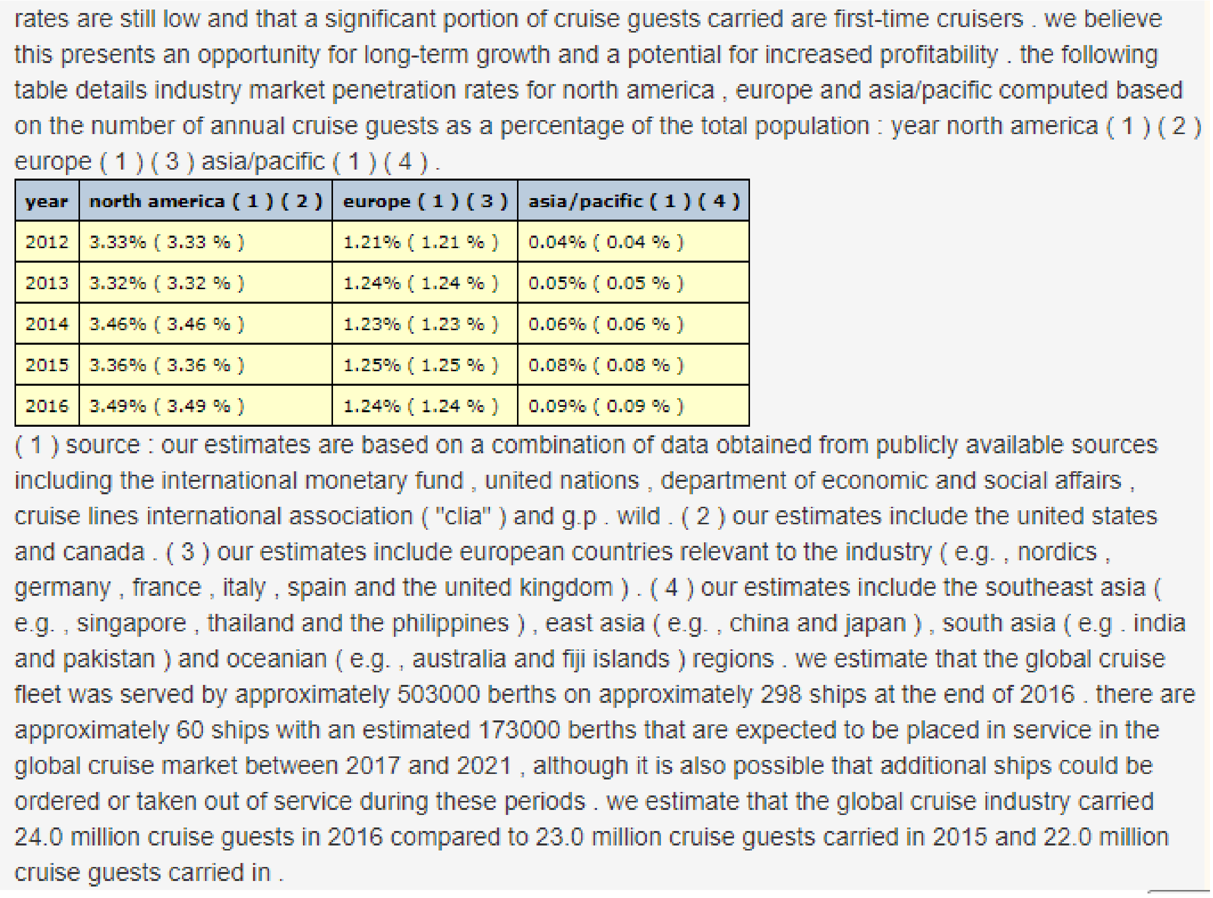
Input to LLMs will be given in the form of prompts. Prompt used for question-answering task will be formatted in such a manner so that LLM can easily extract the context, question and answer during fine-tuning. Instructions will also be added to this prompt to create instruction tuning datasets. Prompt template used for fine-tuning in question-answering task has been given in table 4.3.

Table 4.3 – Prompt template for question-answering task

Prompt template
###Instruction: {Instructions about the question and answering task to the model} ###Context: {Input text on which model will be questioned} ###Question: {Question based on context} ###Answer: {Actual answer}

The first component of the prompt template is the instruction which describes the task. The next component is context. Template has an empty line between instruction and context so that LLM can easily differentiate between instruction and context in the prompt. The context will be constructed based on three columns - pre_text (text before the tabular data), table (tabular data) and post_text (text after tabular data) for question and answering task. Question and answer components of the template will be populated from question and answer column respectively.

Each record in the FinQA dataset represents text from a financial report. Fig. 4.1 (Chen et al., 2021b) shows actual text from financial report for a record in FinQA dataset. However, the data in FinQA dataset is not stored in same format as it is in financial report.



rates are still low and that a significant portion of cruise guests carried are first-time cruisers . we believe this presents an opportunity for long-term growth and a potential for increased profitability . the following table details industry market penetration rates for north america , europe and asia/pacific computed based on the number of annual cruise guests as a percentage of the total population : year north america (1) (2) europe (1) (3) asia/pacific (1) (4) .

year	north america (1) (2)	europe (1) (3)	asia/pacific (1) (4)
2012	3.33% (3.33 %)	1.21% (1.21 %)	0.04% (0.04 %)
2013	3.32% (3.32 %)	1.24% (1.24 %)	0.05% (0.05 %)
2014	3.46% (3.46 %)	1.23% (1.23 %)	0.06% (0.06 %)
2015	3.36% (3.36 %)	1.25% (1.25 %)	0.08% (0.08 %)
2016	3.49% (3.49 %)	1.24% (1.24 %)	0.09% (0.09 %)

(1) source : our estimates are based on a combination of data obtained from publicly available sources including the international monetary fund , united nations , department of economic and social affairs , cruise lines international association ("clia") and g.p . wild . (2) our estimates include the united states and canada . (3) our estimates include european countries relevant to the industry (e.g. , nordics , germany , france , italy , spain and the united kingdom) . (4) our estimates include the southeast asia (e.g. , singapore , thailand and the philippines) , east asia (e.g. , china and japan) , south asia (e.g . india and pakistan) and oceanian (e.g. , australia and fiji islands) regions . we estimate that the global cruise fleet was served by approximately 503000 berths on approximately 298 ships at the end of 2016 . there are approximately 60 ships with an estimated 173000 berths that are expected to be placed in service in the global cruise market between 2017 and 2021 , although it is also possible that additional ships could be ordered or taken out of service during these periods . we estimate that the global cruise industry carried 24.0 million cruise guests in 2016 compared to 23.0 million cruise guests carried in 2015 and 22.0 million cruise guests carried in .

Figure 4.1 – Text from financial report for a sample record in FinQA (Chen et al., 2021b)

The text before tabular data shown in Fig. 4.1 is stored in pre_text column and text after tabular data is stored in post_text column and tabular data is stored in table column. The data in pre_text and post_text columns are stored as list of separate sentences instead of a single paragraph as shown in table 4.4. This will be combined into a single paragraph in this study so that it can be used in the prompt.

Table 4.4 – Pre_text and post_text columns in FinQA dataset

Pre_text	Post_text
["rates are still low and that a significant portion of cruise guests carried are first-time cruisers .", "we believe this presents an opportunity for long-term growth and a potential for increased profitability .", "the following table details industry market penetration rates for north america , europe and asia/pacific computed based on the number of annual cruise guests as a percentage of the total population : year north america (1) (2) europe (1) (3) asia/pacific (1) (4) ."]	["(1) source : our estimates are based on a combination of data obtained from publicly available sources including the international monetary fund , united nations , department of economic and social affairs , cruise lines international association ("clia") and g.p .", "wild ." "(2) our estimates include the united states and canada .", "(3) our estimates include european countries relevant to the industry (e.g. , nordics , germany , france , italy , spain and the united kingdom) .", "(4) our estimates include the southeast asia (e.g. , singapore , thailand and the philippines) , east asia (e.g. , china and japan) , south asia (e.g .", "india and pakistan) and oceanian (e.g. , australia and fiji islands) regions .", "we estimate that the global cruise fleet was served by approximately 503000 berths on approximately 298 ships at the end of 2016 .", "there are approximately 60 ships with an estimated 173000 berths that are expected to be placed in service in the global cruise market between 2017 and 2021 , although it is also possible that additional ships could be ordered or taken out of service during these periods .", "we estimate that the global cruise industry carried 24.0 million cruise guests in 2016 compared to 23.0 million cruise guests carried in 2015 and 22.0 million cruise guests carried in ."]

The tabular data in the financial report is stored as list of lists in the FinQA dataset as shown in table 4.5. Each record in the tabular data is stored as a separate list. This will be converted into a pipe delimited table in this study.

Table 4.5 – Table column in FinQA dataset

Table
[['year', 'north america (1) (2)', 'europe (1) (3)','asia/pacific (1) (4)'], ['2012', '3.33% (3.33 %)', '1.21% (1.21 %)', '0.04% (0.04 %)'], ['2013', '3.32% (3.32 %)', '1.24% (1.24 %)', '0.05% (0.05 %)'], ['2014', '3.46% (3.46 %)', '1.23% (1.23 %)', '0.06% (0.06 %)'], ['2015', '3.36% (3.36 %)', '1.25% (1.25 %)', '0.08% (0.08 %)'], ['2016', '3.49% (3.49 %)', '1.24% (1.24 %)', '0.09% (0.09 %)']]

The necessary transformations have been applied on pre_text, post_text and table columns in the dataset and concatenated with question and answer columns to produce the text in prompt format for fine-tuning. Formatted prompt (equivalent to financial report shown in Fig 4.1) is shown in table 4.6.

Table 4.6 – Formatted prompt for fine-tuning in question-answering task

<table>
<tr><th>Final Prompt</th></tr>
<tr><td>"###Instruction: Financial report data and analysis has been given as context , please respond to the question using the context given.

###Context:
rates are still low and that a significant portion of cruise guests carried are first-time cruisers .we believe this presents an opportunity for long-term growth and a potential for increased profitability .the following table details industry market penetration rates for north america , europe and asia/pacific computed based on the number of annual cruise guests as a percentage of the total population : year north america (1) (2) europe (1) (3) asia/pacific (1) (4) .

year|north america (1) (2)|europe (1) (3)|asia/pacific (1) (4)
2012|3.33% (3.33 %)|1.21% (1.21 %)|0.04% (0.04 %)
2013|3.32% (3.32 %)|1.24% (1.24 %)|0.05% (0.05 %)
2014|3.46% (3.46 %)|1.23% (1.23 %)|0.06% (0.06 %)
2015|3.36% (3.36 %)|1.25% (1.25 %)|0.08% (0.08 %)
2016|3.49% (3.49 %)|1.24% (1.24 %)|0.09% (0.09 %)

(1) source : our estimates are based on a combination of data obtained from publicly available sources including the international monetary fund , united nations , department of economic and social affairs , cruise lines international association (""clia"") and g.p .wild .(2) our estimates include the united states and canada .(3) our estimates include european countries relevant to the industry (e.g. , nordics , germany , france , italy , spain and the united kingdom) .(4) our estimates include the southeast asia (e.g. , singapore , thailand and the philippines) , east asia (e.g. , china and japan) , south asia (e.g .india and pakistan) and oceanian (e.g. , australia and fiji islands) regions .we estimate that the global cruise fleet was served by approximately 503000 berths on approximately 298 ships at the end of 2016 .there are approximately 60 ships with an estimated 173000 berths that are expected to be placed in service in the global cruise market between 2017 and 2021 , although it is also possible that additional ships could be ordered or taken out of service during these periods .we estimate that the global cruise industry carried 24.0 million cruise guests in 2016 compared to 23.0 million cruise guests carried in 2015 and 22.0 million cruise guests carried in .

####question:at the end of 2016, what was the average number of berths per ship in the global cruise fleet?

###Answer1687.92"</td></tr>
</table>

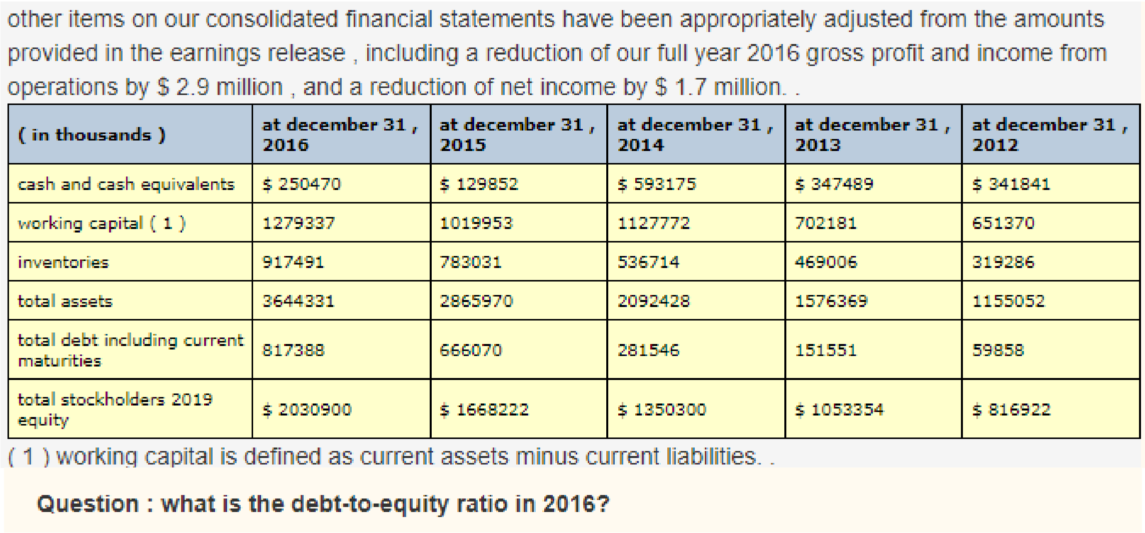
FinQA dataset has been split into train and test by the study which introduced this dataset. Train dataset contains 6251 records and test dataset contains 1147 records. Instruction which describes the task will also be added to the prompt for train dataset. One of the instructions from set of instructions shown in table 4.7 will be selected at random and will be added to the prompt for each record in the dataset.

Table 4.7 – Instructions for FinQA dataset

Instructions - FinQA Dataset
Financial data and expert analysis has been provided as context, use the provided context to answer the question.
Please provide the required answer to this expert-authored finance question based on the context given.
A deep financial question based on a financial report has been given along with required context, could you help answer the financial question.
Analysis based on financial documents has been given as context, please answer the question using the context.
Financial report data and analysis has been given as context , please respond to the question using the context given.

Multiple instructions are used to describe the task so that LLM can capture the essence and fully understand the task. Train dataset which has the final prompt (after necessary transformations have been applied and instructions added) for each record will be used to fine-tune the LLMs (LLaMA-2, Mistral, and Falcon) used in this study. The final prompt template for test dataset will be like train dataset, however, will not contain the instruction (as it is required only for fine-tuning) and the actual answer. The LLMs will be evaluated on test dataset once fine-tuning is completed.

Fig. 4.2 gives an example of context and question from test dataset. Context contains financial statement details in tabular form along with text to describe the tabular data. The question for LLM in this example is to calculate debt to equity ratio for the year 2016. The LLM needs to have capability to extract the required information for the question – total debt (817388) and total equity ($2030900) from the column named – at december 31,2016. LLM also needs to have numerical capability to understand that total debt needs to be divided with total equity to arrive at the answer – 0.402. This example shows that test dataset will evaluate the LLMs on general question answering capabilities (which involve extract the relevant information for a particular question) and numerical reasoning capabilities.



other items on our consolidated financial statements have been appropriately adjusted from the amounts provided in the earnings release , including a reduction of our full year 2016 gross profit and income from operations by $ 2.9 million , and a reduction of net income by $ 1.7 million. .

(in thousands)	at december 31 , 2016	at december 31 , 2015	at december 31 , 2014	at december 31 , 2013	at december 31 , 2012
cash and cash equivalents	$ 250470	$ 129852	$ 593175	$ 347489	$ 341841
working capital (1)	1279337	1019953	1127772	702181	651370
inventories	917491	783031	536714	469006	319286
total assets	3644331	2865970	2092428	1576369	1155052
total debt including current maturities	817388	666070	281546	151551	59858
total stockholders 2019 equity	$ 2030900	$ 1668222	$ 1350300	$ 1053354	$ 816922

(1) working capital is defined as current assets minus current liabilities. .

Question : what is the debt-to-equity ratio in 2016?

Figure 4.2 – Example from FinQA test dataset (Chen et al., 2021b)

Financial phrasebank dataset will be used for sentiment analysis task in this study. This is a benchmark dataset used in multiple studies and it is cleansed and pre-processed by the study which introduced this dataset and hence no additional processing is required. However, data needs to be converted into a format which can used to fine-tune the LLMs. As per the study (Malo et al., 2014) which introduces this dataset, it has text from financial press releases and news which were annotated as positive, negative, or neutral sentiment by experts. The study also mentions that multiple domain experts worked on labelling the sentences and not all experts agreed on sentiment for a particular sentence. Hence, the study released various variants of the dataset. Study mentions that original dataset has 5000 records, however >50% of experts agreed on the sentiment label for 4840 records, >66% of experts agreed on the label for 4211 records, >75% of experts agreed on label for 3448 records and 100% of experts agreed for 2259 records (see Table 4.8).

Table 4.8 - Example from FinQA test dataset (Malo et al., 2014)

Dataset	Negative %	Neutral %	Positive %	Count
Sentences with 100% agreement	13.4	61.4	25.2	2259
Sentences > 75% agreement	12.2	62.1	25.7	3448
Sentences > 66% agreement	12.2	60.1	27.7	4211
Sentences > 50% agreement	12.5	59.4	28.2	4840

This research will consider the dataset where all experts agree on the sentiment label for sentences in the dataset (first row in Table 4.8). This dataset contains 2259 records. This dataset will be split into 75% train dataset and 25% test dataset.

Prompt template used for fine-tuning in sentiment analysis task has been shown in Table 4.9. The prompt for this dataset will contain the instruction for this task, followed by the context which contains the text for which sentiment needs to be predicted. This is followed by question and answer (actual sentiment of input text).

Table 4.9 – Prompt template for Sentiment analysis task

Prompt template
###Instruction: {Instructions about the sentiment analysis task to the model} ###Context: {Input text for which LLM will predict the sentiment} ###Question: {Question to the LLM on sentiment of given input text} ###Answer: {Actual sentiment of the sentence}

Table 4.10 shows the instructions which will be added to prompt in train dataset. Similar to that of question-answering task, one of the instructions will be selected at random and added to the prompt template. Table 4.11 shows the dataset description of financial phrasebank dataset.

Table 4.10 – Instructions for Sentiment analysis task

Instructions – Financial phrasebank Dataset
This statement has been extracted from a financial news article. Please provide your answer as either positive, negative or neutral.
The following financial news sentence has been provided as input. Determine if the sentiment is positive, negative, or neutral?
What is the sentiment of sentence extracted from financial news report? The provided options are positive, negative, or neutral.
Based on the text extracted from the financial news report, determine whether the sentiment is positive, negative, or neutral.
The sentiment of the text extracted from a financial news source needs to be evaluated. The sentiment could be positive, negative, or neutral.
This statement has been extracted from a financial news piece and its sentiment needs to be determined. Indicate whether it is positive, negative, or neutral.

Table 4.11 – Financial phrasebank dataset description

Column	Datatype	Description
sentence	String	Financial text for which sentiment needs to be predicted
label	Integer	Sentiment of the text. 0-Negative, 1 -Neutral and 2-Positive

Fig. 4.3 gives the percentage of records in the dataset for each sentiment label. The dataset has around 62% of records having neutral sentiment, 25% of records as positive sentiment and 13% of records as negative sentiment. This observation is same as (Malo et al., 2014) who introduced this financial phrasebank dataset. Hence, the main interpretation of this chart is that this dataset is representative of finance texts and apt choice for evaluation using sentiment analysis tasks for finance domain.

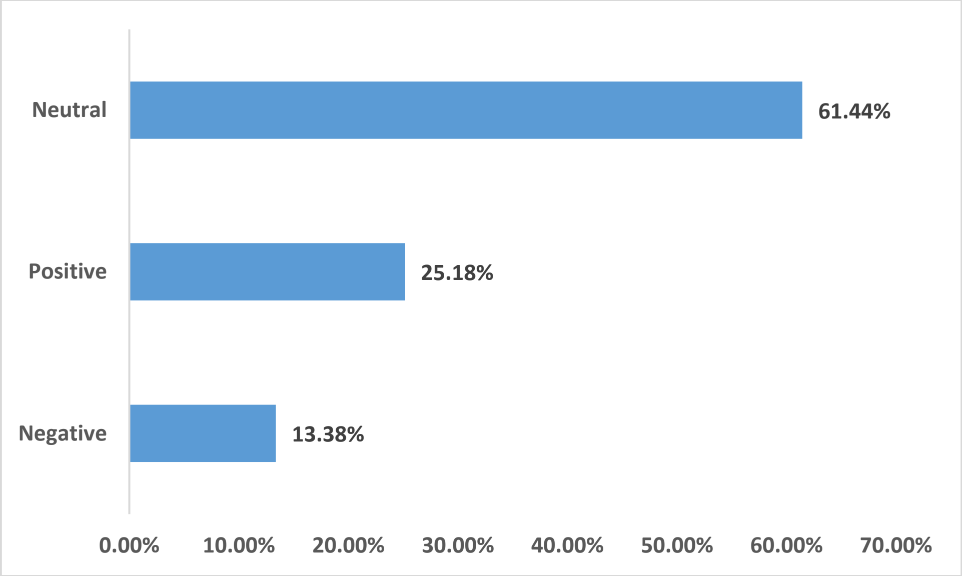


Figure 4.3 – % of records per sentiment label

Table 4.12 shows a sample record for each label from the dataset. The class labels – 0, 1 and 2 will be mapped to negative, neutral and positive before being added as answer in the prompt. The sentence column will be used populate the context in the prompt template. Table 4.13 shows the final prompt for a sample record.

Table 4.12 – Financial phrasebank dataset records

Sentence	Label
For the last quarter of 2010, Componenta 's net sales doubled to EUR131m from EUR76m for the same period a year earlier, while it moved to a zero pre-tax profit from a pre-tax loss of EUR7m.	2 (positive)
The international electronic industry company Elcoteq has laid off tens of employees from its Tallinn facility; contrary to earlier layoffs the company contracted the ranks of its office workers, the daily Postimees reported	0 (Negative)
Technopolis plans to develop in stages an area of no less than 100,000 square meters in order to host companies working in computer technologies and telecommunications, the statement said.	1 (Neutral)

Table 4.13 – Formatted prompt for fine-tuning in sentiment analysis task

Final Prompt
###Instruction: This statement has been extracted from a financial news article. Please provide your answer as either positive, negative, or neutral. ###Context: For the last quarter of 2010, Componenta 's net sales doubled to EUR131m from EUR76m for the same period a year earlier, while it moved to a zero pre-tax profit from a pre-tax loss of EUR7m. ###Question: Determine the sentiment of given input text ###Answer: Positive

This prompt will be used for fine-tuning. For evaluation, answer component of the prompt will be removed. Once, the data has been transformed to produce the final prompts, the next step is to set up the configuration for fine-tuning by selecting the best hyperparameters and this will be discussed in section 4.3.

4.3 Hyperparameters

Identifying the best possible hyperparameters is a crucial part of fine-tuning LLMs. There are two types of hyperparameters for fine-tuning LLMs - LoRA hyperparameters and general neural network hyperparameters. LoRA config determines the number of trainable parameters for fine-tuning and hence it is necessary to determine optimal set of hyperparameters for LoRA. Rank, alpha and choice of transformer weight matrices for enabling LoRA are hyperparameters used to setup the LoRA config.. Learning rate, number of epochs and choice of optimizer are important neural network hyperparameters. Experiments will be performed as part of this research to identify the best possible set of hyperparameters to be used for framework used for fine-tuning LLMs.

4.3.1 LoRA Config Hyperparameters

LoRA rank is the most important hyperparameter from fine-tuning perspective since it determines the number of trainable parameters in the model. The main idea behind LoRA is that weight matrices in the transformers (in LLMs) can be represented by matrices of lower rank (compared to the original rank of weight matrices) without compromising on performance. Choosing a very low rank will reduce the number of trainable parameters resulting in significant reduction in compute resources consumed during fine-tuning. However, a small rank may not work for every task or dataset and hence it is necessary to experiment with different values of rank to determine the best value to be used (Hu et al., 2021). Experiments will be conducted using various values of rank to identify the best rank for each LLM used in this study.

Weight matrix is scaled by (alpha/rank) and hence higher value of alpha assigns more weight to LoRA activations. (Hu et al., 2021) suggests using alpha which equals the rank and hence same will be followed in this study. LoRA can be enabled for query, key, value, or output weight matrices of transformers (in LLMs). (Hu et al., 2021) show that enabling LoRA for query and value weight matrices gives the best performance compared to other combinations. Hence, LoRA will be enabled for query and value weight matrices as part of this research.

4.3.2 Neural Network Hyperparameters

Learning rate is a way of controlling the rate at which model updates the parameters during training. This determines the how fast the model moves towards optimal weights. If the learning rate is a very large value, the optimal solution will be skipped and if it is too small, then it requires too many iterations to converge to best values for the weights. This shows the importance of choosing an optimal learning rate value. The existing literature (Lee et al., 2023) uses 4e-4 as optimal learning rate for fine-tuning LLMs and same will be used in this study. One epoch is defined as a cycle where entire training dataset is passed forward and backward through the neural network and weights are calculated based on the same. Increasing the number of epochs can help the model to refine its understanding, however excessive epochs can lead to overfitting. In conventional deep learning, neural networks are trained using multiple epochs, however experiments on fine-tuning LLMs performed by

(Raschka, 2023) show that multi epoch training leads to performance decline and overfitting. The existing literature on fine-tuning LLMs (Lee et al., 2023) also use one epoch for fine-tuning and hence number of epochs will be set to one in this study. Studies (Choi et al., 2019) show that Adam optimizer is among the best optimizers and hence it will be used in this study. Other important hyperparameters has been summarized in Table 4.14

The values of hyperparameters mentioned in this chapter has been used to fine-tune all three LLMs. Data has been loaded from the datasets and converted into a format which can be used for fine-tuning. This is followed by setting up the fine-tuning framework using the hyperparameters discussed in this chapter. The base models (LLaMA-2, Falcon and Mistral) have been downloaded and loaded from hugging face hub (described in section 4.4). These models were fine-tuned on training data (instruction tuning datasets) using the fine-tuning framework and saved for inference and evaluation. The fine-tuned models were used for detailed evaluation and comparative analysis was performed based on the same (explained in detail in chapter 5 – Results and Discussion).

Table 4.14 – Other important Hyperparameters

Hyperparameter	Description
max_seq_length	This is the maximum sequence length of the input prompt. This has been set to 2000. This number was selected based on size of input text in the dataset.
group_by_length	This parameter groups sequences into batches with same length. This saves memory and speeds up training considerably. Hence, this parameter has been set to true for this study.
bnb_4bit_quant_type	This parameter is used to implement quantization. This involves discretizing the weights and biases parameter values in the neural network and storing them using fewer bits. This parameter will be set to nf4 to implement this and nf4 is normalized floating point data type which quantizes the values to 4 bits.
Learning Rate Scheduler	This is used to adjust the learning rate between epochs during training. This study uses cosine learning rate scheduler which reduces the learning rate using a cosine based schedule. This has been selected since it is one of most widely used schedulers.
gradient_checkpointing	This is a technique which is used to save memory during training of neural networks, and this will be set to true.

4.4 Hardware Configuration and Software Tools and Packages

Google colab platform is the environment which has been used to develop the programming framework for this research. Google colab is a cloud-based platform which can be used to develop and execute python code. The major advantage with google colab is that it provides compute resources (GPUs) required to fine-tune and evaluate LLMs. The free version of google colab provides access to T4 tesla GPU which has GPU memory up to 16GB. Fine-tuning LLMs requires more than 16GB of GPU memory and hence Google Pro plus which is paid version with access to powerful premium GPUs with higher memory has been used in this research. Google colab pro plus provides access to A100 GPU which has been developed by Nvidia for AI applications has 80GB of memory.

The programming framework in this study has been developed in python (version 3.10.12). Table 4.14 provides the detailed information on key libraries used in this study. Hugging face is an open-source platform which hosts the LLMs and has datasets and packages necessary for fine-tuning and evaluating the LLMs considered in this work. Hugging face provides relevant APIs to directly load the LLMs hosted in hugging face platform. The fine-tuned models can also be uploaded to hugging face models hub for inference purposes. Hence hugging face libraries has been used in this study to load the data, perform fine-tuning and evaluation of LLMs.

Table 4.15 – Python libraries and packages used in this study.

Library	Version	Description
Transformers	4.31.0	Transformers is a general purpose library developed by hugging face for accessing and using LLMs. This has tools for loading, tokenisation, inference and evaluation of LLMs. This library will be used to load the LLMs and perform evaluation in this study.
PEFT	0.4.0	PEFT library will be used to define the LoRA configuration which enables reducing the number of trainable parameters.
TRL	0.4.7	TRL is a hugging face library which is specifically focussed on fine-tuning the LLMs. This library will be used to fine-tune the LLMs in this study.
Bits and Bytes	0.40.2	This library will be used to implement QLoRA and create quantized models for faster computation.
Pandas	1.5.3	Pandas is an open-source python library designed for data analysis, transformation and manipulation. This library will be used to format the prompts to produce the final prompt template.
Sci-kit Learn	1.2.2	Sci-kit learn is an open-source python library which provides tools for building models, cross-validation, hyper-parameter tuning and evaluation. This library will be used in this study to calculate the precision, recall and F1-score metrics for sentiment analysis task

4.5 Summary

The first section (section 4.2) of analysis and design chapter gives detailed description of datasets used in this study. This section also delves into transformation steps implemented to convert data into a format which can be used to fine-tune LLMs. This section also gives examples of data before and after the transformation steps. Section 4.3 discusses in detail about the hyperparameters which will be used in this study. The final section (section 4.4) discusses about the hardware configuration which has been used to fine-tune the LLMs. This section also discusses about software tools and packages which has been used in this study.

CHAPTER 5

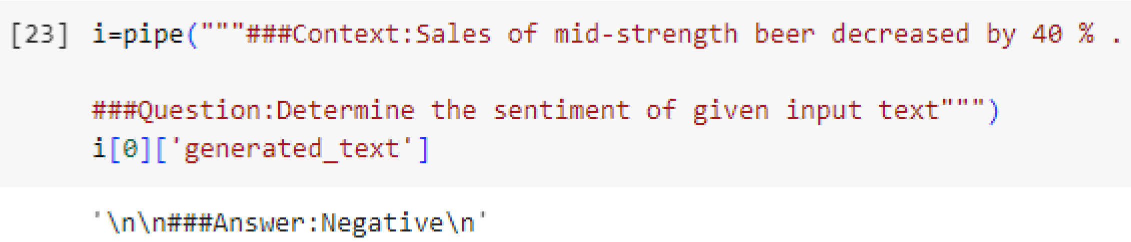
RESULTS AND DISCUSSIONS

5.1 Introduction

The results obtained in this research are discussed in this chapter. All results have been explained in detail along with interpretation. LLaMA-2, Falcon and Mistral LLMs have been considered in this study. These models are trained and evaluated on question-answering and sentiment analysis tasks. Section 5.2 discusses about results obtained for sentiment analysis tasks for all three LLMs considered in this research. Section 5.3 is focussed on results of question-answering tasks. Section 5.4 provides the overall consolidated comparative results of all three LLMs across tasks along with detailed discussion on implications of those results.

5.2 Sentiment analysis results

Evaluation of sentiment analysis has been performed on the test dataset. Precision, recall and F1-score metrics has been calculated for the three fine-tuned LLMs considered in this research. Basic post-processing has been performed to convert the obtained result from the LLMs into a format which can be used for evaluation against sentiment labels in the test dataset. Fig. 5.1 shows the result of a record from test dataset.



```
[23] i=pipe("""###Context:Sales of mid-strength beer decreased by 40 % .

     ###Question:Determine the sentiment of given input text""")
     i[0]['generated_text']

     '\n\n###Answer:Negative\n'
```

Figure 5.1 – Sentiment analysis evaluation result from fine-tuned LLM

The output of the LLM contains certain additional text like newline characters (‘\n’) and ‘###Answer:’ in addition to the actual sentiment. This is because the prompt used for fine-tuning contains the question followed by two new lines and ‘###Answer:’ and then the actual sentiment of the text as shown in Table 5.1. Prompt is designed in this manner to make sure that LLMs can differentiate between the input text and actual sentiment. Hence, during evaluation the LLM produces the output in the same format. The newline characters and

‘###Answer:’ has been removed from the text and resulting output contains only sentiments – positive, negative and neutral for corresponding input text from test dataset. This sentiment value – negative, neutral and positive have been mapped to 0, 1 and 2 respectively so that it can be compared with actual sentiment labels (0, 1 and 2) in the test dataset.

Table 5.1 – Prompt used for fine-tuning LLMs

Prompt
###Instruction: This statement has been extracted from a financial news article. Please provide your answer as either positive, negative or neutral. ###Context: For the last quarter of 2010, Componenta 's net sales doubled to EUR131m from EUR76m for the same period a year earlier , while it moved to a zero pre-tax profit from a pre-tax loss of EUR7m . ###Question: Determine the sentiment of given input text ###Answer: Positive

As discussed in previous chapter rank is an important hyperparameter while fine-tuning LLMs and hence as part of this study, two values of rank i.e., 64 and 128 were considered and experiments were conducted to find the rank with best performance. Table 5.2 shows the comparison of evaluation results for rank 64 and 128 for fine-tuned LLaMA-2 model.

Table 5.2 – Evaluation results for rank 64 and 128

	Rank-64			**Rank-128**		
Sentiment Class	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Negative	0.96	0.96	0.96	0.97	0.93	0.95
Neutral	0.96	0.97	0.97	0.98	0.97	0.97
Positive	0.92	0.94	0.93	0.93	0.98	0.96
	Macro Average F1-Score		**0.96**	**Macro Average F1-Score**		**0.96**

The precision, recall and F1-scores for individual sentiment labels (positive, negative and neutral) with rank 64 is similar to with rank 128. The final macro average F1-score is same in both cases. The interpretation from these results is that both the cases have the same predictive ability and there is not much difference in terms of performance. The number of trainable parameters with rank 128 will be double that of rank 64 and hence will require higher number of compute resources for fine-tuning. Therefore, rank 64 will be used to fine-tune the LLMs in this study to optimize the compute resources without compromising on results.

5.2.1 LLaMA-2 fine-tuned for sentiment analysis task

The fine-tuned LLaMA-2 has been used to generate the predictions on test dataset and evaluation has been carried out. Table 5.3 shows the confusion matrix for negative sentiment label. This confusion matrix has been built based on one vs rest methodology where negative sentiment is considered as one class and other labels (positive and neutral) are considered as other class. Confusion matrices for other sentiments have also been built in the same manner. There are totally 76 records with negative sentiment out of which 73 has been predicted correctly by the model (TP). There are 490 instances of other labels, out of which 487 have been predicted correctly by the model (TN).

Table 5.3 - Confusion matrix for negative sentiment

		Actual	
		Negative Sentiment	**Other Sentiments**
Predicted	**Negative Sentiment**	73 (TP)	3 (FP)
	Other Sentiments	3 (FN)	487 (TN)

Precision, recall and F1-score are calculated based on numbers from confusion matrix and these numbers vary between 0 and 1. To achieve high precision score would require the model to reduce the number of false positives. Precision measures the proportion of predicted positive instances that are actually correct (TP) out of the total number of instances predicted as positive (TP+FP). Recall measures the proportion of actual positives that have been accurately predicted (TP) out of the total actual positives (TP+FN) by the model. F1-score is harmonic mean of Precision and Recall. These calculations are performed as given below:

$$\text{Precision} = TP/(TP+FP) =73/76 = 0.96$$

$$\text{Recall} = TP/(TP+ FN) = 73/76 = 0.96$$

F1-score = 2*(Precision*Recall)/(Precision +Recall) = 0.96

The number of false positive and false negatives are same and hence the precision and recall have the same value of 0.96. Since both precision and recall are the same, F1-score which is harmonic mean of precision and recall is also the same i.e., 0.96. The interpretation from these numbers is that the model has a very good predictive ability in classifying the negative sentiment label.

Table 5.4 shows the confusion matrix for neutral sentiment class. The number of true positives and true negatives are high with very a smaller number of false positives and false negatives. This results in a high F1-score of 0.97. Hence, the interpretation based on these numbers is that LLaMA-2 model has high predictive ability in classifying neutral sentiments. The calculation of precision, recall and F1-score for this case is given below:

Precision = TP/(TP+FP) =335/344 = 0.97

Recall = TP/(TP+ FN) = 335/348 = 0.96

F1-score = 2*(Precision*Recall)/(Precision +Recall) = 0.97

Table 5.4 - Confusion matrix for neutral sentiment

		Actual	
		Neutral Sentiment	**Other Sentiments**
Predicted	**Neutral Sentiment**	335 (TP)	9 (FP)
	Other Sentiments	13 (FN)	209 (TN)

Table 5.5 shows the confusion matrix for positive sentiment class. The confusion matrix numbers for positive sentiment labels shows that the model is capable of accurately classifying the positive sentiment records. This is evident by high F1-score of 0.93. Calculations for precision, recall and F1-score are as given below for this case:

Precision = TP/(TP+FP) =134/146 = 0.92

Recall = TP/(TP+ FN) = 134/142 = 0.94

F1-score = 2*(Precision*Recall)/ (Precision +Recall) = 0.93

Table 5.5 - Confusion matrix for positive sentiment

		Actual	
		Positive Sentiment	**Other Sentiments**
Predicted	**Positive Sentiment**	134 (TP)	12 (FP)
	Other Sentiments	8 (FN)	412 (TN)

Table 5.6 summarises the precision, recall and F1-score numbers across the sentiment labels and also provides the overall aggregated F1-score metrics. Macro average F1-score is the average of F1-scores across sentiment classes.

Table 5.6 - Overall performance of fine-tuned LLaMA-2 model

	LLaMA-2 (Fine-tuned)		
Sentiment Class	**Precision**	**Recall**	**F1-Score**
Negative	0.96	0.96	0.96
Neutral	0.96	0.97	0.97
Positive	0.92	0.94	0.93
	Macro Average F1-Score		**0.96**

The fine-tuned LLaMA-2 model has high Precision, Recall and F1-scores across sentiment classes and macro average F1-score is high as well. The key interpretation based on these high scores is that LLaMA-2 model provides outstanding performance on sentiment analysis task.

5.2.2 Falcon fine-tuned for sentiment analysis task

The fine-tuned Falcon model has been evaluated using the test dataset. Table 5.7 shows the confusion matrix for negative sentiment class. There are very small number of false positives and no false negatives, and this leads to high precision, recall and F1-score. This shows that model can accurately predict the negative sentiment in the dataset.

Calculations of precision, recall and F1-score are shown below:

Precision = TP/(TP+FP) =76/80 = 0.95

Recall = TP/(TP+ FN) = 76/76 = 1

F1-score = 2*(Precision*Recall)/ (Precision +Recall) = 0.97

Table 5.7 – Confusion matrix for negative sentiment

		Actual	
		Negative Sentiment	**Other Sentiments**
Predicted	**Negative Sentiment**	76 (TP)	4 (FP)
	Other Sentiments	0 (FN)	486 (TN)

Table 5.8 shows the confusion matrix for neutral sentiment. Precision, recall and F1-score are 0.99, 0.97 and 0.98 respectively. These values are calculated as shown below:

Precision = TP/(TP+FP) =336/340 = 0.99

Recall = TP/(TP+ FN) = 336/348 = 0.97

F1-score = 2*(Precision*Recall)/ (Precision +Recall) = 0.98

Table 5.8 – Confusion matrix for neutral sentiment

		Actual	
		Neutral Sentiment	**Other Sentiments**
Predicted	**Neutral Sentiment**	336 (TP)	4 (FP)
	Other Sentiments	12 (FN)	214 (TN)

Similar to negative sentiment, number of false positives and false negatives are very small leading to high precision, recall and F1-score. This shows Falcon has high predictive ability in classifying neutral sentiments in the dataset.

Table 5.9 shows the confusion matrix for positive sentiment. Precision, recall and F1-scores are high (see calculations below) showing the ability of the model to accurately classify the positive sentiments in the dataset.

Precision = TP/(TP+FP) =138/146 = 0.95

Recall = TP/(TP+ FN) = 138/142 = 0.97

F1-score = 2*(Precision*Recall)/ (Precision +Recall) = 0.96

Table 5.9 – Confusion matrix for positive sentiment

		Actual	
		Positive Sentiment	**Other Sentiments**
Predicted	**Positive Sentiment**	138 (TP)	8 (FP)
	Other Sentiments	4 (FN)	416 (TN)

Table 5.10 shows the summary of all the metrics for fine-tuned Falcon LLM. The high values of precision, recall and F1-scores across sentiment classes and high macro average F1-score for the model showcases the outstanding performance of fine-tuned Falcon model on sentiment analysis task.

Table 5.10 - Overall performance of fine-tuned Falcon model

	Falcon (Fine-tuned)		
Sentiment Class	**Precision**	**Recall**	**F1-Score**
Negative	0.95	1.00	0.97
Neutral	0.99	0.97	0.98
Positive	0.95	0.97	0.96
	Macro Average F1-Score		**0.97**

5.2.3 Mistral fine-tuned for sentiment analysis task

The fine-tuned Mistral model has been evaluated on the test dataset for sentiment analysis task. Table 5.11 shows the confusion matrix for negative sentiment class. Precision, recall, F1-score is calculated as below using these values. The high value of precision, recall and F1-score shows the ability of fine-tuned Mistral model to accurately classify the negative sentiment in the test dataset.

Precision = TP/(TP+FP) =75/77 = 0.97

Recall = TP/(TP+ FN) = 75/76 = 0.99

F1-score = 2*(Precision*Recall)/ (Precision +Recall) = 0.98

Table 5.11 – Confusion matrix for negative sentiment

		Actual	
		Negative Sentiment	**Other Sentiments**
Predicted	**Negative Sentiment**	75 (TP)	2 (FP)
	Other Sentiments	1 (FN)	488 (TN)

Table 5.12 shows the confusion matrix for neutral sentiment class. The number of false positives and false negatives are less, resulting in high precision, recall and F1-score for neutral sentiment class (see calculations below).

Precision = TP/(TP+FP) =345/356 = 0.97

Recall = TP/(TP+ FN) = 345/348 = 0.99

F1-score = 2*(Precision*Recall)/ (Precision +Recall) = 0.98

Table 5.12 – Confusion matrix for neutral sentiment

		Actual	
		Neutral Sentiment	**Other Sentiments**
Predicted	**Neutral Sentiment**	345 (TP)	11 (FP)
	Other Sentiments	3 (FN)	207 (TN)

Table 5.13 shows the confusion matrix for positive sentiment class. In this case precision, recall and F1-score observed are 0.99, 0.93 and 0.96 respectively. These calculation are shown below:

Precision = TP/(TP+FP) =132/133 = 0.99

Recall = TP/(TP+ FN) = 132/142 = 0.93

F1-score = 2*(Precision*Recall)/ (Precision +Recall) = 0.96

Table 5.13 – Confusion matrix for positive sentiment

		Actual	
		Positive Sentiment	**Other Sentiments**
Predicted	**Positive Sentiment**	132 (TP)	1 (FP)
	Other Sentiments	10 (FN)	423 (TN)

The precision, recall and F1-scores of positive sentiment is also high similar to that of other sentiment classes. Table 5.14 shows the overall performance of fine-tuned Mistral model. The metrics show that Mistral shows good performance in sentiment analysis task.

Table 5.14 - Overall performance of fine-tuned Mistral model

	Mistral (Fine-tuned)		
Sentiment Class	**Precision**	**Recall**	**F1-Score**
Negative	0.97	0.99	0.98
Neutral	0.97	0.99	0.98
Positive	0.99	0.93	0.96
	Macro Average F1-Score		**0.97**

5.2.4 Overall results for sentiment analysis task:

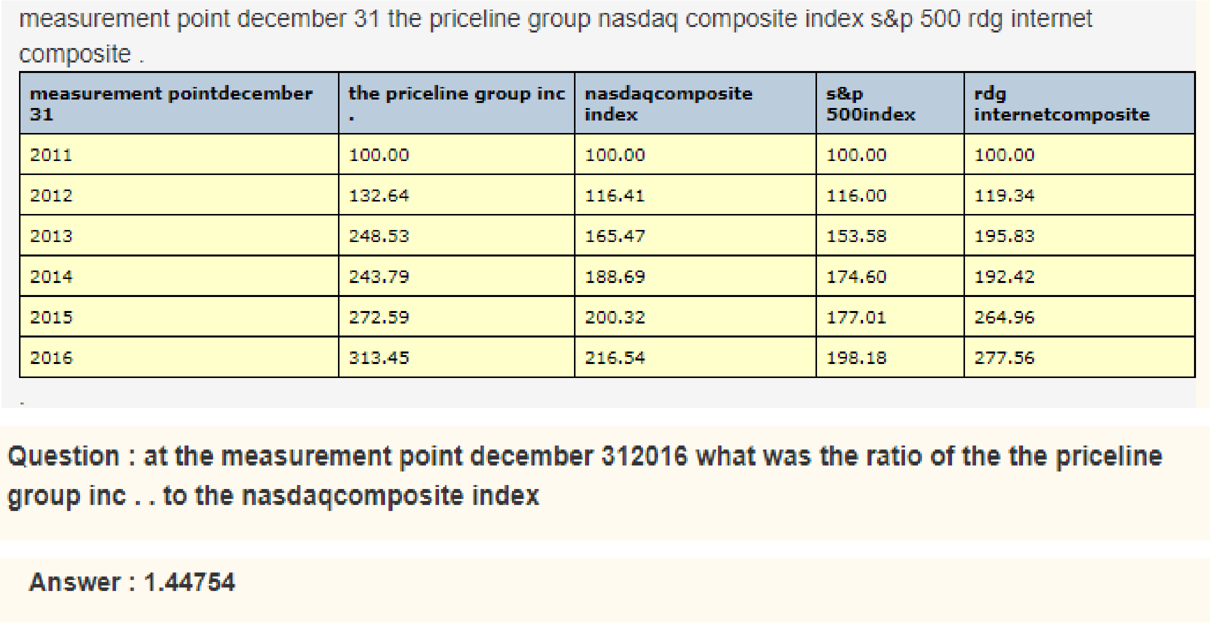
Table 5.15 shows the overall evaluation metrics for all three fine-tuned LLMs – LLaMA-2, Falcon and Mistral. The average of precision, recall and F1-score across sentiment classes (positive, negative, and neutral) for all three LLMs has been calculated. The numbers for all three LLMs are very close to one which indicates ability to predict all the three sentiment classes with a high degree of accuracy. Hence, the key interpretation is that all the three models showcase outstanding performance in sentiment analysis task. Even though, there is not much difference between the numbers, Falcon has slightly higher macro average F1-score compared to other two LLMs and hence it is the best performing model among the three LLMs considered in this research.

Table 5.15 - Overall evaluation metrics for sentiment analysis task

LLM	Macro Average Precision	Macro Average Recall	Macro Average F1-Score
LLaMA-2	0.95	0.96	0.95
Falcon	0.96	0.98	0.97
Mistral	0.98	0.97	0.97

5.3 Question-Answering results

LLaMA-2, Falcon and Mistral LLMs considered for this study have been evaluated on question-answering task using the exact match accuracy metrics. Evaluation has been performed on the test dataset. Similar to sentiment analysis, basic post processing has been performed to convert the output from LLMs into a format which can be used for comparison against actual answers in test dataset. Fig. 5.2 shows sample record from test dataset.



measurement point december 31 the priceline group nasdaq composite index s&p 500 rdg internet composite .

measurement pointdecember 31	the priceline group inc .	nasdaqcomposite index	s&p 500index	rdg internetcomposite
2011	100.00	100.00	100.00	100.00
2012	132.64	116.41	116.00	119.34
2013	248.53	165.47	153.58	195.83
2014	243.79	188.69	174.60	192.42
2015	272.59	200.32	177.01	264.96
2016	313.45	216.54	198.18	277.56

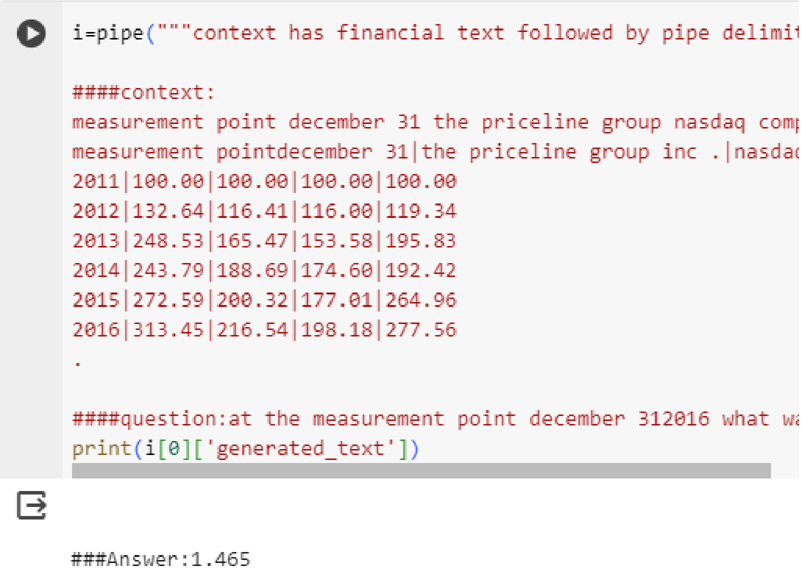
.

Question : at the measurement point december 312016 what was the ratio of the the priceline group inc . . to the nasdaqcomposite index

Answer : 1.44754

Figure 5.2 – Record from test dataset

Fig. 5.3 shows the result for this dataset from fine-tuned Mistral LLM. The actual answer – 1.44754 is almost equal to the predicted answer (1.465), however for exact match accuracy the predicted values must be an exact match of actual answers, hence both the predicted and actual answers are rounded off to nearest integer and compared to check if it is matching. This is done because question-answering task in finance domain is generally performed on reports which contains investment research and analysis. The decimals do not have any major impact in this type of reports and hence numbers are rounded-off.



```
i=pipe("""context has financial text followed by pipe delimi

####context:
measurement point december 31 the priceline group nasdaq com
measurement pointdecember 31|the priceline group inc .|nasda
2011|100.00|100.00|100.00|100.00
2012|132.64|116.41|116.00|119.34
2013|248.53|165.47|153.58|195.83
2014|243.79|188.69|174.60|192.42
2015|272.59|200.32|177.01|264.96
2016|313.45|216.54|198.18|277.56
.

####question:at the measurement point december 312016 what w
print(i[0]['generated_text'])
```

```
###Answer:1.465
```

Figure 5.3 – Result from Mistral LLM for record from test dataset

Similar to sentiment analysis, newline characters and words like ‘###Answer:’ are removed from the LLM output to extract the numerical answer. The actual answers in test dataset are numerical in nature and post processing is performed to remove any non-numeric characters. This enables to extract the numerical answers from LLM to compare with actual answers from test dataset.

Exact match accuracy scores checks if predicted value matches with actual value. If output matches exactly with the record of the dataset, then score is 1, else it is 0. The exact match scores for all records are summed up and divided by total number of records to get the exact match score for the model. The final scores vary between 0 and 1 and if the number is close to one it indicates the model is performing well on question answering task.

Exact match accuracy = Records where predictions exactly match answers/Total number of records in the dataset.

Similar to sentiment analysis task, performance of LLaMA-2 model with rank 64 was compared with model having rank as 128. Exact match accuracy of model with rank 128 was

0.16 and it was 0.15 for model with rank 64. Rank 64 has been used as hyperparameter while fine-tuning to optimize the computational resources since there is no huge difference in performance between rank 128 and rank 64 scores.

Table 5.16 shows the results of question-answering task for all three fine-tuned LLMs considered in this study. The evaluation results show that exact match accuracy numbers are quite low for all three LLMs. However, another key interpretation is that Mistral was able to outperform the other models (LLaMA-2 and Falcon) by a significant margin. The exact match accuracy of Mistral is more than two times that of other LLMs considered in this study. These results show that there is still significant scope for improvement for open-source LLMs in question-answering tasks based on numerical reasoning for finance domain.

Table 5.16 - Overall evaluation results for question-answering task

Model	Total No. of Records	No. of Records where Answers match	Exact Match Accuracy
LLaMA-2	1147	172	0.15
Falcon	1147	138	0.12
Mistral	1147	379	0.33

5.4 Discussion on overall results

Three LLMs considered for this study were evaluated using appropriate evaluation metrics on sentiment analysis task as well as question-answering task. The overall results across these tasks have been summarized in Table 5.17.

Table 5.17 – Overall results

	Sentiment Analysis	Question-Answering
Model	**Macro Average F1-Score**	**Exact Match Accuracy**
LLaMA-2	0.95	0.15
Falcon	0.97	0.12
Mistral	**0.96**	**0.33**

All three LLMs showed outstanding performance on sentiment analysis task, however the performance was much lower for question-answering task. This shows that open-source LLMs have much scope for improvement in question-answering task based on numerical reasoning. Even though the scores are lower in question-answering task, score of Mistral is more than double of other two models' scores. This shows that Mistral has much better numerical reasoning capabilities compared to other models. Mistral was the best model in question-answering task and performance was equivalent to other models in sentiment analysis task. Hence, the key interpretation based on these results is that Mistral is the best performing model among the three models considered in this research. This finding correlates with finding of the study (Jiang et al., 2023) which introduces the Mistral LLM. This study mentions that Mistral outperforms LLaMA-2 and other equivalent open-source LLMs on general tasks. The findings of this study shows that this holds true for finance domain as well. The technical and functional stakeholders in finance domain can take these results into consideration when deciding the best model for their use cases and applications. As mentioned in aims and objectives, LLMs have been fine-tuned and thorough comparative analysis has been performed and results have been reported based on the same.

5.5 Summary

The evaluation of sentiment analysis task and question-answering task has been discussed in detail in this chapter. The results for sentiment analysis have been explained with help of confusion matrix for each sentiment class. The overall macro average F1-score was also reported for each model. The exact match accuracy was calculated and reported for question-answering task. The overall results for all three models across two tasks have been reported to identify the best performing model.

CHAPTER 6

CONCLUSIONS AND FUTURE RECOMMENDATIONS

6.1 Introduction

This chapter provides summary of all the research carried out and results obtained as part of this study. This chapter also focusses on the limitations and future recommendations based on the observations during this research.

6.2 Summary of research and findings

Problem statement and background of this research has been established in the introduction chapter. Research gaps were identified based on thorough literature review and objectives were framed based on these research gaps.

This research is focussed on fine-tuning three LLMs i.e., LLaMa-2, Falcon and Mistral on finance domain data and providing a detailed comparative analysis through evaluation. Hence, key NLP tasks used for performance evaluation were selected through literature review. Literature review revealed that sentiment analysis and question-answer tasks are important in finance domain. Similarly, evaluation metrics i.e., precision, recall and F1-score was also selected based on literature review for sentiment analysis whereas exact match accuracy was used for question-answer task. Literature review has also been performed on fine-tuning techniques of LLMs considered in this study. Research states LoRA technique can reduce the number of trainable parameters by a great extent without compromising on the performance. Literature also states that this results in significant reduction in amount of compute resources consumed. Hence, this technique has been used in this study to fine-tune the LLMs. Literature on LoRA technique also states that rank is an important hyperparameter which needs to be selected through experimentation. Hence, experiments were conducted in this study to select the optimal value of rank for fine-tuning LLMs using LoRA technique. FinQA and Financial phrasebank datasets were used in this study for question-answering and sentiment analysis tasks respectively. These datasets are benchmark datasets in the finance domain for these tasks and hence have been used in this study. End-to-end programming framework was developed to load the data, convert it into a format which can be used to fine-tune the LLM, perform the fine-tuning and evaluation. The models were fine-tuned using the fine-tuning framework and evaluation was done on the test dataset using appropriate evaluation metrics. The confusion matrix metrics, precision, recall and F1-score was reported for each sentiment class for each model. The overall macro average F1-score was also reported for each model. The exact match accuracy was reported for question-answering task for all three LLMs.

As per the results, all three LLMs show outstanding performance (F1 scores above 0.94) on sentiment analysis , however, the performance was lower on question-answering task (exact match accuracy of 0.15 for LLaMA 2, 0.12 for Falcon and 0.33 for Mistral). However, Mistral scored more than double the scores of other two LLMs on question-answering. As per overall analysis of the results, the interpretation is that Mistral performs shows outstanding performance on sentiment analysis (equivalent to that of other two LLMs) and clearly outperforms LLaMA 2 and Falcon on question-answering task. Hence, Mistral is the best performing LLM among the LLMs considered in this research. The findings of this study can be used by functional and technical stakeholders in finance domain when selecting LLMs for their use-case or application.

The findings of the research with respect to the aims and objectives stated has been listed.

To build comprehensive end-to-end programming framework to fine-tune the LLMs.

Programming framework to load the data, transform it into a format which can be used for fine-tuning and perform the fine-tuning of LLMs using optimal hyperparameters has been built. Detailed evaluation of LLMs was also done using the programming framework developed as part of this study. Google colab platform was used in this study to develop the programming framework since it gives access to high end premium GPUs which are required for fine-tuning and evaluating LLMs. The hugging face libraries and packages were used to load the datasets and fine-tune the LLMs.

To create instruction tuning datasets by overlaying the instructions (prompts) on train dataset.

The train data was converted into a format which can be used to fine-tune the LLMs, and instructions were added to it to create instruction tuning datasets. These datasets were used to fine-tune the LLMs.

To fine-tune the LLMs on instruction tuning datasets.

All three LLMs considered in this study have been fine-tuned on finance domain data and saved for inference and evaluation.

To evaluate the fine-tuned models on NLP tasks relevant to finance domain.

The fine-tuned models were evaluated using appropriate evaluation metrics to perform a thorough and comprehensive comparative analysis. LLMs were evaluated on Sentiment analysis task using precision, recall and F1-score metrics. Exact match accuracy was used to evaluate the LLMs on question-answering task. The overall results show that Mistral is best performing LLM among the three LLMs considered for this study.

6.3 Limitations of the study and future recommendations

There are other larger variants of LLaMA-2 and Falcon models which were not added to scope of this study due to constrains on time, compute and finances required to fine-tune the LLMs. Future studies can explore the large variants of these models by fine-tuning them on finance data for various other tasks and carry out similar comparative analysis. The result of this research shows that performance of open-source LLMs on numerical-reasoning based on finance data was lower. Research can be conducted in future to improve the numerical reasoning capabilities of open-source LLMs especially in the context of finance domain. The other open-source LLMs not considered in this research can also be explored. Future studies can explore other open-source LLMs (which were not in scope of this study) by fine-tuning them on finance data for various other tasks and carry out similar comparative analysis.

REFERENCES:

Aghajanyan, A., Yu, L., Conneau, A., Hsu, W.-N., Hambardzumyan, K., Zhang, S., Roller, S., Goyal, N., Levy, O. and Zettlemoyer, L., (2023) Scaling Laws for Generative Mixed-Modal Language Models. [online] Available at: http://arxiv.org/abs/2301.03728.

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F. and Sanghai, S., (2023) GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. [online] Available at: http://arxiv.org/abs/2305.13245.

Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever, (2018) *Improving Language Understanding by Generative Pre-Training*. [online] Available at: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf [Accessed 17 Aug. 2023].

Alghazo, J.M., Kazmi, Z. and Latif, G., (2017) Cyber security analysis of internet banking in emerging countries: User and bank perspectives. In: *2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS)*. [online] IEEE, pp.1–6. Available at: http://ieeexplore.ieee.org/document/8277910/.

Al-Hawari, F. and Barham, H., (2021) A machine learning based help desk system for IT service management. *Journal of King Saud University - Computer and Information Sciences*, [online] 336, pp.702–718. Available at: https://linkinghub.elsevier.com/retrieve/pii/S1319157819300515.

Arbane, M., Benlamri, R., Brik, Y. and Alahmar, A.D., (2023) Social media-based COVID-19 sentiment classification model using Bi-LSTM. *Expert Systems with Applications*, [online] 212, p.118710. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0957417422017353.

Bahdanau, D., Cho, K. and Bengio, Y., (2014) Neural Machine Translation by Jointly Learning to Align and Translate. [online] Available at: http://arxiv.org/abs/1409.0473.

Bandari, V., (2023) Enterprise Data Security Measures: A Comparative Review of Effectiveness and Risks Across Different Industries and Organization Types. *International Journal of Business Intelligence and Big Data Analytics*, [online] 61, pp.1–11. Available at: https://research.tensorgate.org/index.php/IJBIBDA/article/view/3 [Accessed 26 Nov. 2023].

Beltagy, I., Peters, M.E. and Cohan, A., (2020) Longformer: The Long-Document Transformer. [online] Available at: http://arxiv.org/abs/2004.05150.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., Mccandlish, S., Radford, A., Sutskever, I. and Amodei, D., (2020a) Language Models are Few-Shot Learners. [online] Available at: https://papers.nips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf [Accessed 17 Aug. 2023].

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., (2020b) Language Models are Few-Shot Learners. [online] Available at: http://arxiv.org/abs/2005.14165.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y.T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M.T. and Zhang, Y., (2023) Sparks of Artificial General Intelligence: Early experiments with GPT-4. [online] Available at: http://arxiv.org/abs/2303.12712.

Chen, Z., Chen, W., Smiley, C., Shah, S., Borova, I., Langdon, D., Moussa, R., Beane, M., Huang, T.-H., Routledge, B. and Wang, W.Y., (2021a) FinQA: A Dataset of Numerical Reasoning over Financial Data. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. [online] Stroudsburg, PA, USA: Association for Computational Linguistics, pp.3697–3711. Available at: https://aclanthology.org/2021.emnlp-main.300 [Accessed 8 Aug. 2023].

Chen, Z., Li, S. and Smiley, C., (2021b) *FinQA Dataset Official website*. [online] Available at: https://finqasite.github.io/explore.html [Accessed 25 Nov. 2023].

Chen, Z., Li, S., Smiley, C., Ma, Z., Shah, S. and Wang, W.Y., (2022) ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering. [online] Available at: http://arxiv.org/abs/2210.03849.

Chiang, W.-L., Li, Z., Lin, Z. and Sheng, Y., (2023) *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality*. [online] Available at: https://lmsys.org/blog/2023-03-30-vicuna/ [Accessed 22 Nov. 2023].

Choi, D., Shallue, C.J., Nado, Z., Lee, J., Maddison, C.J. and Dahl, G.E., (2019) On Empirical Comparisons of Optimizers for Deep Learning. [online] Available at: http://arxiv.org/abs/1910.05446.

Clark, K., Luong, M.-T., Le, Q. V. and Manning, C.D., (2020) ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. [online] Available at: http://arxiv.org/abs/2003.10555.

Curme, C. and Stanley, H.E., (2015) Coupled Network Approach To Predictability Of Financial Market Returns And News Sentiments. *International Journal of Theoretical and Applied Finance*, [online] 1807, p.1550043. Available at: https://www.worldscientific.com/doi/abs/10.1142/S0219024915500430.

Dettmers, T., Pagnoni, A., Holtzman, A. and Zettlemoyer, L., (2023) QLoRA: Efficient Finetuning of Quantized LLMs. [online] Available at: http://arxiv.org/abs/2305.14314.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K., (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North*. [online] Stroudsburg, PA, USA: Association for Computational Linguistics, pp.4171–4186. Available at: http://aclweb.org/anthology/N19-1423.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., Yi, J., Zhao, W., Wang, X., Liu, Z., Zheng, H.-T., Chen, J., Liu, Y., Tang, J., Li, J. and Sun, M., (2023) Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, [online] 53, pp.220–235. Available at: https://www.nature.com/articles/s42256-023-00626-4.

Ghiassi, M., Skinner, J. and Zimbra, D., (2013) Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications*, [online] 4016, pp.6266–6282. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0957417413003552.

Greff, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R. and Schmidhuber, J., (2017) LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, [online] 2810, pp.2222–2232. Available at: http://ieeexplore.ieee.org/document/7508408/.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D. and Smith, N.A., (2020) Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. [online] Stroudsburg, PA, USA: Association for Computational Linguistics, pp.8342–8360. Available at: https://www.aclweb.org/anthology/2020.acl-main.740.

Hong Hui Tan and King Hann Lim, (2019) Vanishing Gradient Mitigation with Deep Learning Neural Network Optimization. In: *2019 7th International Conference on Smart Computing & Communications (ICSCC)*. [online] pp.1–4. Available at: https://ieeexplore.ieee.org/document/8843652 [Accessed 21 Oct. 2023].

Houlsby, N., Giurgiu, A., Jastrze¸bski, S.J., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M. and Gelly, S., (2019a) Parameter-Efficient Transfer Learning for NLP. [online] Available at: https://proceedings.mlr.press/v97/houlsby19a/houlsby19a.pdf [Accessed 22 Nov. 2023].

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M. and Gelly, S., (2019b) *Parameter-Efficient Transfer Learning for NLP*. [online] Available at: http://arxiv.org/abs/1902.00751.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W., (2021) LoRA: Low-Rank Adaptation of Large Language Models. [online] Available at: http://arxiv.org/abs/2106.09685.

Huang, A., Zang, A., Zheng, R., Berger, P., Buslepp, W., Chen, K., Cheng, Q., Core, J., Dechow, P., Dichev, I., Li, X., Lin, L., Mikhail, M., Seasholes, M., You, H. and Zimmerman, J., (2014) *Evidence on the Information Content of Text in Analyst Reports*. [online] Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1888724 [Accessed 26 Nov. 2023].

Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D. de las, Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.-A., Stock, P., Scao, T. Le, Lavril, T., Wang, T., Lacroix, T. and Sayed, W. El, (2023) Mistral 7B. [online] Available at: http://arxiv.org/abs/2310.06825.

Lee, A.N., Hunter, C.J. and Ruiz, N., (2023) Platypus: Quick, Cheap, and Powerful Refinement of LLMs. [online] Available at: http://arxiv.org/abs/2308.07317.

Leippold, M., (2023) Sentiment spin: Attacking financial sentiment with GPT-3. *Finance Research Letters*, [online] 55, p.103957. Available at: https://linkinghub.elsevier.com/retrieve/pii/S154461232300329X.

Li, C., Ye, W. and Zhao, Y., (2022a) *FinMath: Injecting a Tree-structured Solver for Question Answering over Financial Reports*. [online] Available at: https://aclanthology.org/2022.lrec-1.661/ [Accessed 26 Nov. 2023].

Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J., (2022b) A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and*

Learning Systems, [online] 3312, pp.6999–7019. Available at: https://ieeexplore.ieee.org/document/9451544/.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. and Alsaadi, F.E., (2017) A survey of deep neural network architectures and their applications. *Neurocomputing*, [online] 234, pp.11–26. Available at: https://www.sciencedirect.com/science/article/pii/S0925231216315533 [Accessed 24 Oct. 2023].

Loughran, T. and Mcdonald, B., (2011) When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *The Journal of Finance*, [online] 661, pp.35–65. Available at: https://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.2010.01625.x.

Malo, P., Sinha, A., Korhonen, P., Wallenius, J. and Takala, P., (2014) Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, [online] 654, pp.782–796. Available at: https://onlinelibrary.wiley.com/doi/10.1002/asi.23062.

Mavi, V., Saparov, A. and Zhao, C., (2023) Retrieval-Augmented Chain-of-Thought in Semi-structured Domains. [online] Available at: http://arxiv.org/abs/2310.14435.

Mikolov, T., Chen, K., Corrado, G. and Dean, J., (2013) Efficient Estimation of Word Representations in Vector Space. [online] Available at: http://arxiv.org/abs/1301.3781.

Min, B., Ross, H., Sulem, E., Veyseh, A.P. Ben, Nguyen, T.H., Sainz, O., Agirre, E., Heintz, I. and Roth, D., (2024) Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey. *ACM Computing Surveys*, [online] 562, pp.1–40. Available at: https://dl.acm.org/doi/10.1145/3605943 [Accessed 25 Oct. 2023].

Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L.T. and Trajanov, D., (2020) Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers. *IEEE Access*, [online] 8, pp.131662–131682. Available at: https://ieeexplore.ieee.org/document/9142175/.

M.Tarwani, K. and Edem, S., (2017) Survey on Recurrent Neural Network in Natural Language Processing. *International Journal of Engineering Trends and Technology*, [online] 486, pp.301–304. Available at: http://www.ijettjournal.org/archive/ijett-v48p253.

OpenAI, (2023) GPT-4 Technical Report. [online] Available at: http://arxiv.org/abs/2303.08774.
Otter, D.W., Medina, J.R. and Kalita, J.K., (2021) A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, [online] 322, pp.604–624. Available at: https://ieeexplore.ieee.org/document/9075398/.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J. and Lowe, R., (2022) Training language models to follow instructions with human feedback. [online] Available at: http://arxiv.org/abs/2203.02155.

Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E. and Launay, J., (2023) The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. [online] Available at: http://arxiv.org/abs/2306.01116.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., (2019) *Language Models are Unsupervised Multitask Learners*. [online] Available at: https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf [Accessed 17 Aug. 2023].

Raschka, S., (2023) *Practical Tips for Finetuning LLMs Using LoRA (Low-Rank Adaptation)*. [online] Available at: https://magazine.sebastianraschka.com/p/practical-tips-for-finetuning-llms [Accessed 30 Nov. 2023].

Rozière, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X.E., Adi, Y., Liu, J., Remez, T., Rapin, J., Kozhevnikov, A., Evtimov, I., Bitton, J., Bhatt, M., Ferrer, C.C., Grattafiori, A., Xiong, W., Défossez, A., Copet, J., Azhar, F., Touvron, H., Martin, L., Usunier, N., Scialom, T. and Synnaeve, G., (2023) Code LLaMA: Open Foundation Models for Code. [online] Available at: http://arxiv.org/abs/2308.12950.

Schick, T., Schütze, H. and Schütze, S., (2022) True Few-Shot Learning with Prompts-A Real-World Perspective. *MIT Press Direct*, [online] 10, pp.716–731. Available at: http://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00485/2030692/tacl_a_00485.pdf.

Schmidhuber, J., (2015) Deep learning in neural networks: An overview. *Neural Networks*, [online] 61, pp.85–117. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0893608014002135.

Shah, R., Chawla, K., Eidnani, D., Shah, A., Du, W., Chava, S., Raman, N., Smiley, C., Chen, J. and Yang, D., (2022) When FLUE Meets FLANG: Benchmarks and Large Pretrained Language Model for Financial Domain. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. [online] Stroudsburg, PA, USA: Association for Computational Linguistics, pp.2322–2335. Available at: https://aclanthology.org/2022.emnlp-main.148.

Shang, L., Xi, H., Hua, J., Tang, H. and Zhou, J., (2023) A Lexicon Enhanced Collaborative Network for targeted financial sentiment analysis. *Information Processing & Management*, [online] 602, p.103187. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0306457322002886.

Shazeer, N., (2020) GLU Variants Improve Transformer. [online] Available at: http://arxiv.org/abs/2002.05202.

Souma, W., Vodenska, I. and Aoyama, H., (2019) Enhanced news sentiment analysis using deep learning methods. *Journal of Computational Social Science*, [online] 21, pp.33–46. Available at: http://link.springer.com/10.1007/s42001-019-00035-x.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B. and Liu, Y., (2021) RoFormer: Enhanced Transformer with Rotary Position Embedding. [online] Available at: http://arxiv.org/abs/2104.09864.

Sutskever, I., Vinyals, O. and Le, Q. V., (2014) Sequence to Sequence Learning with Neural Networks. [online] Available at: http://arxiv.org/abs/1409.3215.

Tabassum, A. and Patil, R.R., (2020) A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing. *International Research Journal of Engineering and Technology*, [online] 76, pp.4684–4687. Available at: https://www.irjet.net/archives/V7/i6/IRJET-V7I6913.pdf [Accessed 26 Nov. 2023].

Taori, R., Gulrajani, I. and Zhang, T., (2023) *Alpaca: A Strong, Replicable Instruction-Following Model*. [online] Available at: https://crfm.stanford.edu/2023/03/13/alpaca.html [Accessed 28 Oct. 2023].

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E. and Lample, G., (2023a) *LLaMA: Open and Efficient Foundation Language Models*. [online] Available at: https://arxiv.org/abs/2302.13971 [Accessed 15 Aug. 2023].

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C.C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P.S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E.M., Subramanian, R., Tan, X.E., Tang, B., Taylor, R., Williams, A., Kuan, J.X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S. and Scialom, T., (2023b) LLaMA 2: Open Foundation and Fine-Tuned Chat Models. [online] Available at: http://arxiv.org/abs/2307.09288.

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., (2017) *Attention Is All You Need*. [online] Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf [Accessed 15 Aug. 2023].

Wang, G., Wang, T., Wang, B., Sambasivan, D., Zhang, Z., Zheng, H. and Zhao, B.Y., (2015) Crowds on Wall Street. In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. [online] New York, NY, USA: ACM, pp.17–30. Available at: https://dl.acm.org/doi/10.1145/2675133.2675144.

Wang, Y., Zhao, Y. and Petzold, L., (2023) Are Large Language Models Ready for Healthcare? A Comparative Study on Clinical Language Understanding. [online] Available at: http://arxiv.org/abs/2304.05368.

Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M. and Le, Q. V., (2021) Finetuned Language Models Are Zero-Shot Learners. [online] Available at: http://arxiv.org/abs/2109.01652.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E.H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J. and Fedus, W., (2022) Emergent Abilities of Large Language Models. [online] Available at: http://arxiv.org/abs/2206.07682.

Wenzek, G., Lachaux, M.-A., Conneau, A., Chaudhary, V., Guzmán, F., Joulin, A. and Grave, E., (2019) CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. [online] Available at: http://arxiv.org/abs/1911.00359.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D. and Mann, G., (2023a) BloombergGPT: A Large Language Model for Finance. [online] Available at: http://arxiv.org/abs/2303.17564.

Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q.-L. and Tang, Y., (2023b) A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development. *IEEE/CAA Journal of Automatica Sinica*, [online] 105, pp.1122–1136. Available at: https://ieeexplore.ieee.org/document/10113601/.

Yang, Y., UY, M.C.S. and Huang, A., (2020a) FinBERT: A Pretrained Language Model for Financial Communications. [online] Available at: http://arxiv.org/abs/2006.08097.

Yang, Y., UY, M.C.S. and Huang, A., (2020b) FinBERT: A Pretrained Language Model for Financial Communications. [online] Available at: http://arxiv.org/abs/2006.08097.

Ye, J.F., Chen, X.F., Xu, N.F., Zu, C.F., Shao, Z., Liu, S.F., Cui, Y.F., Zhou, Z.F., Gong, C.F., Shen, Y.F., Zhou, J.F., Chen, S., Gui, T., Zhang, Q.F. and Huang F, X.F., (2023) *A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models*. [online] Available at: https://platform.openai.com/docs/model-index-for-researchers.

Zamfirescu-Pereira, J.D., Wong, R.Y., Hartmann, B. and Yang, Q., (2023) Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. [online] New York, NY, USA: ACM, pp.1–21. Available at: https://dl.acm.org/doi/10.1145/3544548.3581388.

Zhang, B. and Sennrich, R., (2019) Root Mean Square Layer Normalization. [online] Available at: http://arxiv.org/abs/1910.07467.

Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y. and Wen, J.-R., (2023a) A Survey of Large Language Models. [online] Available at: http://arxiv.org/abs/2303.18223.

Zhao, Y., Liu, H., Long, Y., Zhang, R., Zhao, C. and Cohan, A., (2023b) KnowledgeMath: Knowledge-Intensive Math Word Problem Solving in Finance Domains. [online] Available at: http://arxiv.org/abs/2311.09797.

Zhong, R., Lee, K., Zhang, Z. and Klein, D., (2021) Adapting Language Models for Zero-shot Learning by Meta-tuning on Dataset and Prompt Collections. [online] Available at: http://arxiv.org/abs/2104.04670.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., Peng, H., Li, J., Wu, J., Liu, Z., Xie, P., Xiong, C., Pei, J., Yu, P.S. and Sun, L., (2023) A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT. [online] Available at: http://arxiv.org/abs/2302.09419.

Zhu, F., Lei, W., Huang, Y., Wang, C., Zhang, S., Lv, J., Feng, F. and Chua, T.-S., (2021a) *TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance*. [online] Available at: https://arxiv.org/abs/2105.07624 [Accessed 26 Nov. 2023].

Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S. and Chua, T.-S., (2021b) Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering. [online] Available at: http://arxiv.org/abs/2101.00774.

Zhu, Y., Wichers, N., Lin, C.-C., Wang, X., Chen, T., Shu, L., Lu, H., Liu, C., Luo, L., Chen, J. and Meng, L., (2023) SiRA: Sparse Mixture of Low Rank Adaptation. [online] Available at: http://arxiv.org/abs/2311.09179.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. and He, Q., (2019) A Comprehensive Survey on Transfer Learning. [online] Available at: http://arxiv.org/abs/1911.02685.

Zouhar, V., Meister, C., Gastaldi, J.L., Du, L., Vieira, T., Sachan, M. and Cotterell, R., (2023) A Formal Perspective on Byte-Pair Encoding. [online] Available at: http://arxiv.org/abs/2306.16837.

APPENDIX A

RESEARCH PROPOSAL

COMPARATIVE ANALYSIS OF OPEN-SOURCE LARGE LANGUAGE MODELS FOR FINANCE DOMAIN THROUGH IMPLEMENTATION

VIJAYSHANKAR SUBRAMANIAN

Research Proposal

AUGUST 2023

ABSTRACT

Commercial Large Language Models (LLMs) like ChatGPT has disrupted the field of Natural Language Processing (NLP) by showcasing outstanding performance in multiple NLP tasks like question answering, machine translation, text summarization and many others. However, there are some data security and privacy concerns around using commercial LLMs since they require the user queries to be sent to their server for processing. Enterprises are skeptical to use commercial LLMs due to this. Open-source LLMs can help address this challenge since they can be hosted in enterprises' own infrastructure. Also, no additional license cost is involved in using open-source LLMs. For NLP tasks based on a specific domain (like finance), research shows that LLMs fine-tuned on domain specific data improves the performance of LLMs. There are many open-source LLMs available. Therefore, it is necessary to fine-tune the LLMs and perform comparative analysis to identify the strengths and weaknesses of a LLM for a specific domain like finance. LLMs are a recent development in the field of NLP. State-of-the-art research on LLMs for finance domain is focussed on training or fine-tuning a specific LLM using finance domain data. To the best of author's knowledge, there are no other studies which fine-tune multiple open-source LLMs on financial data and provide comparative analysis of LLMs for finance domain. Therefore, main aim of this study is to address this gap by considering five popular open-source LLMs (LLaMA, Pythia, Dolly, Falcon and Vicuna) and fine-tuning all of them on financial data and evaluating them on important NLP tasks (question answering, sentiment analysis and named entity recognition) using appropriate evaluation metrics and provide a thorough and comprehensive comparative analysis. The results and insights from this study will enable the finance domain stakeholders to choose appropriate LLM for their use case.

Table of Contents

List of Abbreviations

LLM…………..	Large Language Model
NLP…................	Natural Language Processing
GPT……………	Generative Pre-trained Transformer
ICL…………….	In-context Learning
LLaMA………..	Large Language Model Meta Artificial Intelligence
BERT………….	Bidirectional Encoder Representations from Transformers
RoBERTa……..	Robustly optimised BERT approach
PEFT………….	Parameter Efficient Fine Tuning
QLoRA……….	Quantized Low Rank Adapters
NER…………..	Named Entity Recognition
CNN………….	Convolutional Neural Network

1. Background

Large Language Model (LLM) is a deep learning model which has been trained on vast amounts of text data from multiple sources like websites, books and various other forms of text content. GPT (Generative Pre-trained Transformer) models are LLMs which are based on transformer architecture (Vaswani et al., 2017). GPT-1(Alec Radford et al., 2018) was the first model in GPT series of LLMs and it was built by using a hybrid approach of unsupervised pre-training and supervised fine-tuning. This model laid the foundation for GPT series models and established the underlying principle to model natural language text which is predicting the next word. GPT-2 (Radford et al., 2019) had similar architecture as GPT-1 but number of parameters was increased to 1.5 billion and model was trained on larger dataset.

Introduction of GPT-3 (Brown et al., 2020a) by OpenAI was a major turning point in development of LLMs. GPT-3 increased number of parameters to 175 billion and introduces in-context learning (ICL). ICL can instruct LLMs to understand tasks in the form of natural language text. GPT-3 shows excellent performance in a variety of Natural Language Processing (NLP) tasks like answering user questions, translation, perform summarisation and generating text. GPT-3 also performs well on tasks that require reasoning abilities and domain adaptation. GPT-3 has been used as foundation model to develop even more capable LLMs. GPT-3, however had some limitations around tasks like completing code and solving math problems and hence codex (Chen et al., 2021a), a GPT model fine-tuned on GitHub code was released by OpenAI. Codex was used to build GitHub co-pilot which can take a programming problem described in natural language as input and generate solution code for the same.

GPT 3.5 was derived from GPT 3 and changes were made on top of GPT 3 to remove toxic output and align the output with user’s intention. ChatGPT, a GPT model optimized for dialogue was built based on GPT 3.5. ChatGPT became one of fastest growing consumer applications in history with around 100 million users monthly. Apart from general LLMs, there are domain specific LLMs like BioGPT (for medicine) and BloombergGPT (for finance) which are trained on vast amounts of data in specific domain. The GPT models developed by OpenAI are commercial LLMs and hence code and dataset used to train the model was not available in public domain. Commercial LLMs have shown great potential by excelling in variety of tasks(Brown et al., 2020b), however they have some crucial limitations. Commercial LLMs require the user queries and data to be sent to their server for processing, in addition to this, some LLMs like ChatGPT save user prompts to train and improve its models. It is also possible some users may inadvertently enter sensitive or personally identifiable information (PII) when specifying the query and this will be sent to servers of commercial LLMs.

Enterprises are skeptical to use commercial LLMs due to above mentioned privacy and security concerns. Open-source LLMs can address the aforementioned challenges that comes with using commercial LLMs for enterprises. Enterprises can use open-source LLMs to deploy these models on their own infrastructure (on-premises or private cloud environment) ensuring sensitive information remains within the enterprise network and thereby reducing the risk of data breaches. Apart from improved security, another major advantage is that there is no additional cost involved in using open-source LLM since it is freely available and hence very useful for enterprises with limited budget.

LLaMA (Large Language Model Meta AI) developed by Meta (Touvron et al., 2023) is one of the most important open-source LLM since it is the foundation model for many other open-source LLMs. There are many other open-source LLMs which have been developed. Alpaca, Vicuna, Bloom, OPT, GLM, Falcon, Pythia, Dolly, GPT-J and Galactica are some of them. Research (Gururangan et al., 2020) shows that language models fine-tuned on data for a specific domain performs better on that tasks on that domain compared to language models which have not been fine-tuned. There are many open-source LLMs available and hence comparative analysis is required to identify the LLM which is best suited for a specific domain like finance. The field of open-source LLMs is relatively new and existing studies regarding LLMs for finance domain are related to training or fine-tuning a specific LLM using finance domain data (Wu et al., 2023) . To the best of author's knowledge, there are no other studies which fine-tune multiple open-source LLMs on financial data and present comparative analysis on the same, therefore, this study aims to bridge that gap by considering five popular (Zhao et al., 2023) open-source LLMs - LLaMA, Pythia, Dolly, Falcon and Vicuna and fine-tuning all of them on financial data and evaluating them on important NLP tasks (question answering, sentiment analysis and named entity recognition) using appropriate evaluation metrics and provide a thorough and comprehensive comparative analysis. Stakeholders in both functional and technical capacities in finance domain can use the results of this study to choose the appropriate LLM for their use case.

2. Related Research

NLP has been a key driver in advancement of finance technology (FinTech) by enabling multiple capabilities from stock price forecasting to advanced financial analytics. Introduction of LLMs has been the most exciting and promising development in field of NLP and they have shown remarkable performance in variety of NLP tasks (Brown et al., 2020b). While the performance of LLMs in general NLP tasks has been outstanding, studies have been shown that language models fine-tuned on domain data perform better on domain specific tasks. (Gururangan et al., 2020) considers data from four domains – news, reviews, biomedical and computer science publications and pre-trained language model known as RoBERTa (Robustly Optimized Bidirectional Encoder Representations from Transformers Approach) is fine-tuned on data from above domains. Evaluation is performed using eight tasks (two in each domain) and results of this study show that the models which have been fine-tuned on domain specific data outperform the models which have not been fine-tuned.

2.1 Language models in finance domain

FinBERT (Finance Bidirectional Encoder Representations from Transformers Approach) is one of the first pre-trained language models to be fine-tuned on financial data. The study (Yang et al., 2020) takes BERT - Bidirectional Encoder Representations from Transformers Approach (Devlin et al., 2019), an existing pre-trained language model and performs fine-tuning on vast amount of financial communication data including earning call transcripts, analyst and corporate reports. Evaluation is done by comparing the accuracy of FinBERT with generic BERT model on three financial sentiment detection datasets and study reports that FinBERT is able to outperform BERT. FLANG (Shah et al., 2022) introduces two pre-trained models fine-tuned on financial data – FLANG-BERT and FLANG-ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately). The base models for these two models are BERT and ELECTRA respectively(Clark et al., 2020). FLANG aims to improve upon FinBERT by employing preferential token masking in masked language models during training. The main objective of masked language models (MLM) is to predict the masked token in a particular sentence. Usually, tokens are masked with equal probability. However, in preferential token masking, preference is given to certain tokens. FLANG models are reported to have outperformed generic BERT, ELECTRA and FinBERT in different tasks like question answering, sentiment analysis and named entity recognition by this study. Introduction of BloombergGPT (Wu et al., 2023) marked a major milestone in development of LLMs for finance. BloombergGPT is a 50 billion parameter LLM which has been trained on a mix of vast amounts of Bloomberg finance data and public data. The base LLM for BloombergGPT is BLOOM (Big Science Large Open-science Open-access Multilingual Language Model). BloombergGPT is evaluated on both finance related and general tasks, both Bloomberg datasets and publicly available finance datasets are utilized for evaluation on finance related tasks. This study reports that BloombergGPT is able to provide superior performance on finance related tasks compared to other models, whereas on general tasks, it is reported to be on par with other models. Despite the outstanding performance BloombergGPT on finance tasks, there are some limitations which makes it difficult for finance enterprises to adopt it. Data security is very important for banking and other financial institutions since they hold large amount of sensitive data. BloombergGPT is a commercial LLM which requires the user queries to be sent to its server for processing and hence this is one of the limitations in its adoption by finance enterprises. Cost involved in using BloombergGPT is another limitation. Open-source LLMs offer a viable alternative to BloombergGPT since enterprises can host them in their own infrastructure and there is no additional cost involved in using open-source LLMs. Finance enterprises can use one of many open-source LLMs by fine tuning them on finance datasets to improve the performance on finance related tasks.

There are many open-source LLMs available that can be adopted by finance domain enterprises. However, there are no comparative studies available which can help finance organisations to take the decision on adopting best available open-source LLM. Therefore, this work aims to present comparative analysis on the same. This work intends to adopt five popular open-source LLM’s i.e., LLaMA, Pythia, Dolly, Falcon and Vicuna. The work aims to fine-tune all of them on financial data and evaluating them on important NLP tasks using appropriate evaluation metrics and provide a thorough and comprehensive comparative analysis. Instruction tuning will be used to fine-tune the LLMs. This is a type of supervised fine-tuning, in which, a collection of instructions and corresponding examples of how to follow the instructions are used to train the LLMs. Instruction tuning datasets will be created by overlaying the instructions (prompts) on the datasets used for this study, these instruction tuning datasets will be used to fine-tune the LLMs. Instruction tuning has been shown to improve performance of LLMs (Wei et al., 2021)

2.2 Overview of Open-Source LLMs

The main aim of this study is to fine tune and compare the performance of open-source LLMs - LLaMA, Pythia, Dolly, Falcon and Vicuna on finance domain data. These are the most popular and widely used open-source LLMs and hence have been used for evaluation in this study.

2.2.1 LLaMA

LLaMA (Touvron et al., 2023) introduced by Meta is one of major open-source LLMs available. It is a collection of LLMs ranging from 7 billion to 65 billion parameters. Publicly available datasets from sources such as Wikipedia, GitHub and StackExchange are used to train the model. Model architecture is similar to that of GPT-3. Evaluation is done on standard LLM evaluation tasks like common sense reasoning, question answering, reading comprehension, mathematical reasoning and code generation. Evaluation is done in both zero-shot (users ask queries to LLMs without providing any context or examples in the query) and few-shot (users provide some context about the query being asked) settings. Study concludes that LLaMA model with 13 billion parameters is able to provide superior performance compared to GPT-3 and similar LLMs.

2.2.2 Vicuna

Vicuna is a LLM which takes LLaMA as foundation model and fine-tunes it using user shared conversations from ShareGPT (a google chrome extension which allows users share their ChatGPT conversations). Vicuna uses instruction tuning technique using above conversations dataset to fine-tune LLaMA.

2.2.3 Pythia

Pythia(Biderman et al., 2023) is a collection of 16 LLMs (starting from 70 million parameters to 12 billion parameters) which has been trained on public data. The model architecture is similar to that of GPT-3.

2.2.4 Dolly

Dolly is a LLM which uses pythia as base model and fine-tunes it using instruction tuning technique with crowd-sourced data from databricks employees.

2.2.5 Falcon

Falcon (Penedo et al., 2023) is a 40 billion parameter LLM which has been trained on public datasets. Falcon utilizes 75% of computational resources used for training GPT-3 but is able to deliver better performance compared to GPT-3. Falcon is also able to match the performance of other state of the art LLMs.

2.3 Model Fine-Tuning and Evaluation

LLMs typically have billions of parameters and fine-tuning these models on a specific dataset would mean updating these parameters and this can be very expensive from a computational perspective. Parameter efficient fine tuning (PEFT) provides an alternative in which only small proportion of parameters in the LLMs are fine-tuned. This significantly reduces the computational resources required for the fine-tuning process. This is also known as delta tuning. The study (Ding et al., 2023) which introduces PEFT shows that it can achieve comparable results to fine-tuning all the parameters of LLM. QLoRA (Quantized Low Rank Adapters) is another technique which can be used to optimize the process of fine-tuning. This technique involves optimizing the memory used to fine-tune the LLMs. PEFT and QLoRA can be used together to fine-tune the LLMs in an optimal and efficient manner.

Datasets considered for fine-tuning in this work are explained in research methodology (Section 6.2). After fine-tuning, LLMs will be evaluated on multiple NLP tasks to perform a comparative analysis. In the past, studies have been performed on comparative analysis (Wang et al., 2023) of LLMs typically used 3 to 5 tasks. On similar lines, the present work considers following three tasks for evaluation of the models - question answering based on numerical reasoning, sentiment analysis and named entity recognition. These tasks have been chosen based on importance of these tasks to finance domain (details provided in Section 6.4.1) and also timeframe allocated for this study. Evaluation metrics for each task will vary since each task is different and details for the same is given in research methodology (Section 6.4.3).

3. Aims and Objectives

The main aim of this research is to perform a thorough and comprehensive comparative analysis of open-source LLMs for finance domain by fine-tuning on finance data and evaluating the LLMs on multiple NLP tasks.

The research objectives based on above mentioned aim are as follows:

- To create instruction tuning datasets by overlaying the instructions (prompts) on train dataset.
- To build comprehensive end-to-end programming framework to fine-tune the LLMs.
- To fine-tune the LLMs on instruction tuning datasets.
- To evaluate the fine-tuned models on multiple NLP tasks using appropriate evaluation metrics.

4. Significance of the Study

This work is of significance to technical and functional stakeholders in finance who are interested in choosing the best suited open-source LLM as per the requirements for their use case. There are multiple open-source LLMs available and comparative analysis is required to identify the best suited LLM. The results and insights from comparative analysis of open-source LLMs done in this study will provide them with necessary details and metrics required to make the right decision.

5. Scope of the Study

This study considers five popular open-source LLMs - LLaMA, Pythia, Dolly, Falcon and Vicuna. These LLMs are considered since they are the most popular and widely used. There are other open-source LLMs which are not in the scope of this study and can form the basis for future work.

Evaluation of LLMs is done on three NLP tasks - question answering, sentiment analysis and named entity recognition. These NLP tasks have been chosen based on two factors – first is importance of these tasks for financial domain and second is timeframe allocated for this study. Details of reasons for choosing these tasks are given in research methodology (Section 6.1). There are other NLP tasks on which evaluation of LLM could be done but those are out of scope of this study and can be considered for future work.

6. Research Methodology

Methodology used in this research involves following steps:

- Loading the required datasets (FinQA, Financial Phrasebank and FiNER) for respective tasks (question answering, sentiment analysis and named entity recognition).
- Splitting them into train and test datasets.
- Constructing instruction tuned datasets by overlaying the prompts (instructions) on train dataset.
- Building the framework required to fine-tune the model.
- Fine-tuning the selected open-source LLMs on train dataset.
- Evaluating the models using relevant metrics for respective tasks to provide a comprehensive comparative analysis of LLMs for finance domain.

The below flowchart depicts high-level overview of steps involved in this research (section for each step is also mentioned in the flowchart).

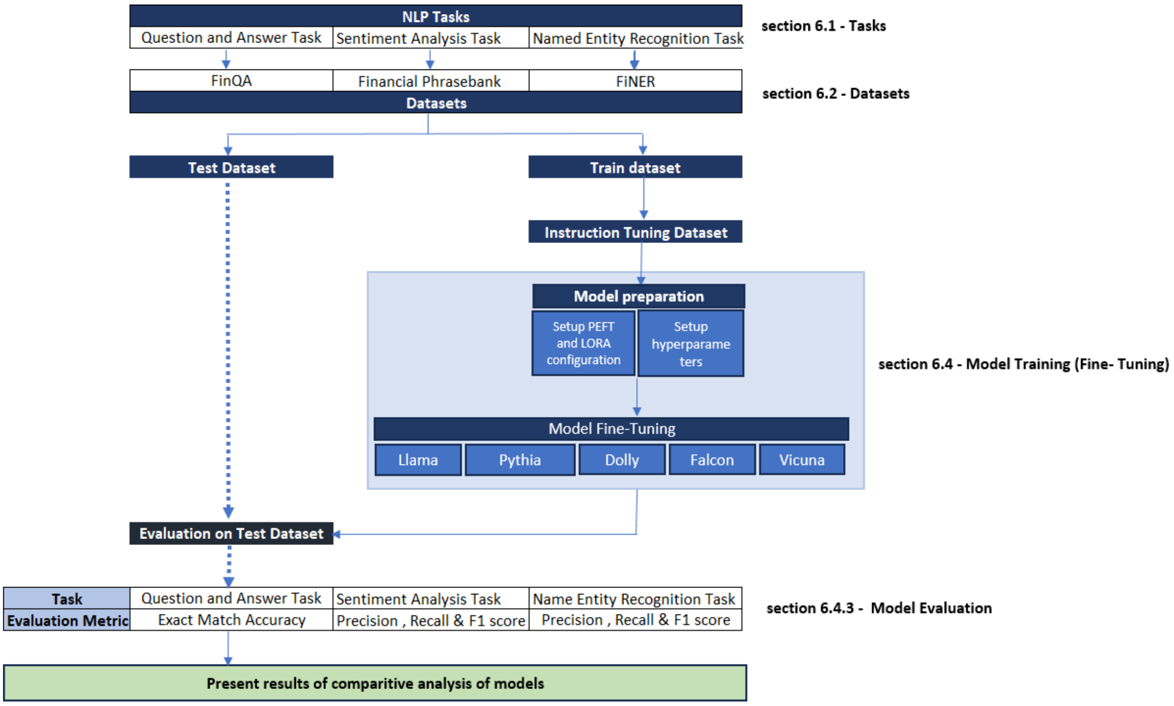


Fig. 1 - Research methodology flowchart

6.1 NLP Tasks

This research involves evaluating the LLMs on multiple NLP tasks since performance of a LLM cannot be determined based on single task. This can help to identify the strengths and weaknesses of each LLM (Wang et al., 2023). Appropriate datasets for each of these tasks will be chosen and those datasets are divided into train and test datasets. LLMs are fine-tuned on train dataset and evaluated on test dataset. NLP tasks considered as part of this study are listed below:

6.1.1 Question-Answering Task for Numerical Reasoning

Question answering task based on numerical reasoning has been chosen because numerical reasoning capability is very important for a LLM which is required to answer queries on financial domain. LLM needs to be able to understand the context of the query from user, retrieve the necessary information and perform numerical calculations wherever required. For example, if a LLM was given profit and loss statement of a company as input and user asked a question - ‘what is the % increase of profit of the company this year compared to last year’. The LLM would need to retrieve the profit of the company this year and last year from the source document and calculate the % increase in profit and return the result. This requires numerical reasoning capability and hence this task is crucial for finance domain.

6.1.2 Sentiment Analysis Task

Sentiment analysis task involves predicting if the given text belongs to positive, negative or neutral sentiment. Sentiment analysis is another important task from financial domain perspective because detecting sentiments is crucial for stock market related use-cases (Song et al., 2017).

6.1.3 Named Entity Recognition Task

Named entity recognition task is also critical because LLM needs to have capabilities to recognize entities (like organisations, persons and location) accurately to understand the context of the query and retrieve the correct information required.

6.2 Datasets

Research involves multiple NLP tasks. There is no single dataset which can be used for different tasks and hence comparative studies of LLMs in this work use different dataset for each task (Wang et al., 2023). Brief information on each dataset used in this work for each NLP tasks is given below:

6.2.1 FinQA dataset for Question-and-Answering Task

Dataset: https://huggingface.co/datasets/dreamerdeo/finqa

(Chen et al., 2021). has released this dataset as part of their study to evaluate financial numerical reasoning abilities of pre-trained models. FinQA dataset consists of 8,821 question answer pairs created based on financial reports, dataset contains both text and tabular data and questions are framed in a manner which requires numerical reasoning to arrive at the answers. Dataset has columns like pre-text and post-text and these columns have data for text before and after the tabular data. Tabular data is present in the column named table. Question column has the numerical questions based on text and tabular data (example – what percent of share repurchases were in fourth quarter). Answer column has the actual answer. (Sun et al., 2022) has used this dataset to improve on retriever generator framework for long form numerical reasoning from RoBERTa model. This Framework uses number-aware negative sampling strategy for retriever to discriminate key numerical facts, and consistency-based reinforcement learning with target program augmentation for the generator to increase the execution accuracy.

6.2.2 Financial Phrasebank dataset for Sentiment Analysis Task

Dataset: https://huggingface.co/datasets/financial_phrasebank

(Malo et al., 2014) has released this dataset as part of their study which explores how semantic orientations can be better detected in financial and economic news. Financial Phrasebank dataset consists of 4850 sentences from financial press releases and news which were tagged as positive, negative or neutral. LLMs will be fine-tuned (trained) using sentences and sentiment labels from train dataset and evaluation will be done using sentences from test dataset. (Adhikari et al., 2023)has used this dataset to propose a new framework based on convolutional neural network (CNN) which improves the explainability of sentiment analysis model.

6.2.3 FiNER dataset for Named Entity Recognition Task

Dataset: https://huggingface.co/datasets/gtfintechlab/finer-ord

(Shah et al., 2023) introduced this dataset for evaluation of language models on Name Entity Recognition task. FiNER dataset comprises of 201 financial news articles with labels for person (PER), location (LOC) and organization (ORG) entities. There is one row for each word (token) in the dataset and it is labelled as either of above entities. LLMs will be fine-tuned using the tokens and corresponding entity labels and followed by evaluation using sentences from test dataset.

6.3 Data Pre-processing

Data pre-processing step is not required in this research as data is clean and it can be used directly to create instruction tuning datasets which will be used for fine-tuning the custom LLMs considered in this work.

6.4 Model Training (Fine-Tuning)

Five open-source LLMs (Zhao et al., 2023) - LLaMA, Pythia, Dolly, Falcon and Vicuna have been considered for this research. (Wei et al., 2021) shows that fine tuning LLMs using instructions (instruction tuning) results in improved performance of LLMs. Five models need to be fine-tuned and evaluated as part of this research and hence it is necessary to follow a modular approach. Generalised and re-usable modules will be created as per requirement which allows multiple models to be fine-tuned and evaluated instead of writing separate code for each model. The individual modules will be integrated to create the final framework. Hugging face transformer libraries will be used to implement the fine-tuning process. Three main modules will be developed as part of this research.

6.4.1 Data Ingestion Module

Data for respective tasks are loaded and split into train and test datasets. Train Datasets for all three tasks – question answering, sentiment analysis and name entity recognition are overlaid with instructions to create instruction tuning datasets. This dataset will be used to fine-tune the models.

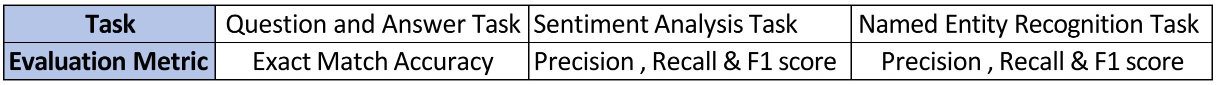
6.4.2 Model Preparation and Fine-tuning module

PEFT and QLoRA are techniques used to efficiently fine-tune the LLMs. PEFT and QLoRA can be used together where PEFT can reduce the number of parameters that needs to be trained and QLoRA can computational complexity of training those parameters. This can result in significant reduction in training time and memory requirements. PEFT and QLoRA configurations and Hyperparameters and tokenizers are setup as part of model preparation.

6.4.3 Evaluation module

There are multiple NLP tasks on which models will be evaluated and hence evaluation metrics relevant to respective task will be used. The below table summarizes the evaluation metric which will be used for each task.

Table 1 - Evaluation metric details



Task	Question and Answer Task	Sentiment Analysis Task	Named Entity Recognition Task
Evaluation Metric	Exact Match Accuracy	Precision , Recall & F1 score	Precision , Recall & F1 score

6.4.3.1 Question-Answering Task

Exact match accuracy is used to evaluate question and answering task(Shao et al., 2023) as numerical reasoning capability of models is being evaluated as part of this task. Therefore, value predicted by the model is compared against actual value and accuracy is calculated.

6.4.3.2 Sentiment Analysis Task

The objective of sentiment analysis task is to predict the sentiment of a given text (positive, negative or neutral). Precision, recall and F1 score will be used as evaluation metric. Studies on sentiment analysis(Arbane et al., 2023) use precision, recall and F1 score as evaluation metric. F1 score takes into account both precision and recall and it is robust to class imbalance and gives an accurate picture of model's performance.

6.4.3.3 Named Entity Recognition Task

NER task involves predicting the person, location and organization entities and predicted entity value is compared against actual value. Evaluation metrics appropriate for this task are precision, recall and F1 score. Studies on NER task (Baigang and Yi, 2023) use F1 score as evaluation metric since it balances precision and recall. Precision checks if the models are predicting many entities which are not named entities and recall checks if the model is missing any of named entities. F1 score combines both precision and recall and hence gives a balanced view of model performance.

The results of evaluation will be collated to provide a comprehensive comparison of multiple LLMs.

7. Resources Requirements

Hardware and software resources required for this study are listed below:

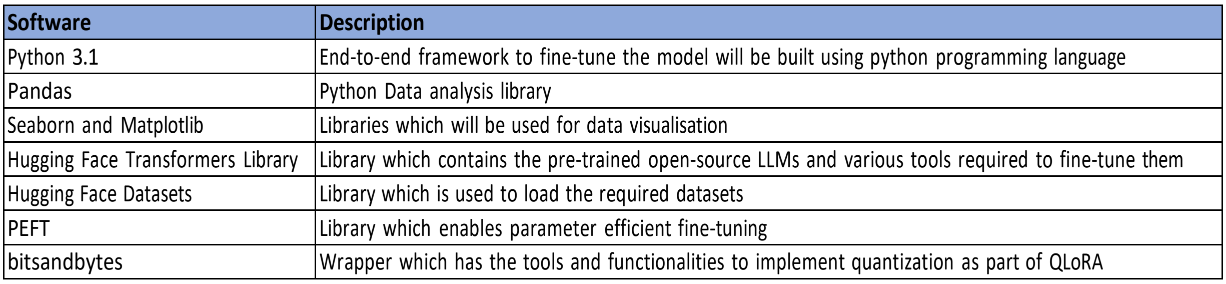
7.1 Hardware resources

Google Colab will be used to build the programming framework required to fine-tune and evaluate the LLMs. Therefore, the default Google Colab configuration of Intel Xeon CPU with 2 vCPUs (virtual CPUs) and 13GB of RAM will be used. This configuration can be upgraded if required.

7.2 Software resources

Software resources which will be used as part of the study have been given below

Table 2 – Software resources



Software	Description
Python 3.1	End-to-end framework to fine-tune the model will be built using python programming language
Pandas	Python Data analysis library
Seaborn and Matplotlib	Libraries which will be used for data visualisation
Hugging Face Transformers Library	Library which contains the pre-trained open-source LLMs and various tools required to fine-tune them
Hugging Face Datasets	Library which is used to load the required datasets
PEFT	Library which enables parameter efficient fine-tuning
bitsandbytes	Wrapper which has the tools and functionalities to implement quantization as part of QLoRA

8. Research Plan

Research plan which will be followed for this study have been given below. Timeframe allocated for development of each module and its sub-steps has been provided. Time required for literature review and preparing the interim report and final thesis report has also been factored into the plan. Start date for this plan has been considered as 24th August (day after research proposal submission). End date of this plan is 6th December.

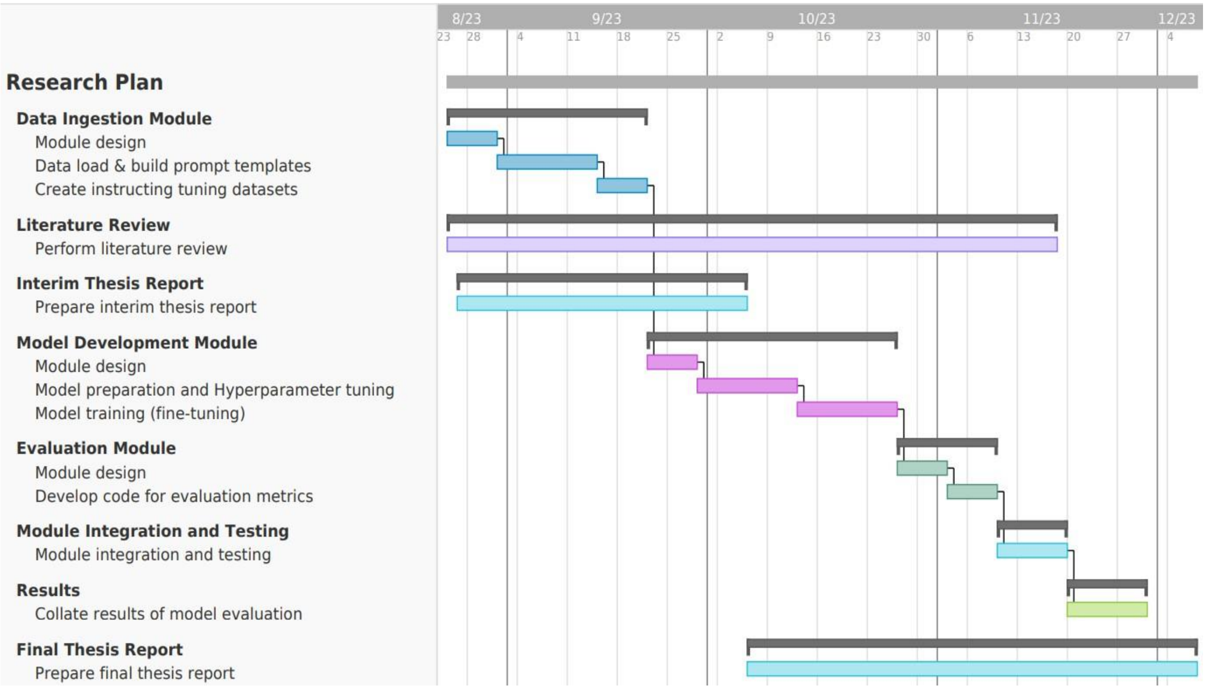


Fig 2 - Research Plan Gantt Chart

8.1 Risk Mitigation

LLMs will be fine-tuned on default configuration provided by Google Colab, there could be a possibility that default configuration may not be sufficient to fine-tune some of the LLMs. This risk can be mitigated by upgrading to higher configuration in Google Colab.

References

Adhikari, S., Thapa, S., Naseem, U., Lu, H.Y., Bharathy, G. and Prasad, M., (2023) Explainable hybrid word representations for sentiment analysis of financial news. *Neural Networks*, [online] 164, pp.115–123. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0893608023001880 [Accessed 8 Aug. 2023].
Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever, (2018) *Improving Language Understanding by Generative Pre-Training*. [online] Available at: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf [Accessed 17
Aug. 2023].

Arbane, M., Benlamri, R., Brik, Y. and Alahmar, A.D., (2023) Social media-based COVID-19 sentiment classification model using Bi-LSTM. *Expert Systems with Applications*, [online] 212, p.118710. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0957417422017353.

Baigang, M. and Yi, F., (2023) A review: development of named entity recognition (NER) technology for aeronautical information intelligence. *Artificial Intelligence Review*, [online] 562, pp.1515–1542. Available at: https://link.springer.com/10.1007/s10462-022-10197-2.

Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M.A., Purohit, S., Prashanth, U.S., Raff, E., Skowron, A., Sutawika, L. and van der Wal, O., (2023) Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. [online] Available at: http://arxiv.org/abs/2304.01373.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., Mccandlish, S., Radford, A., Sutskever, I. and Amodei, D., (2020a) Language Models are Few-Shot Learners. [online] Available at: https://papers.nips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
[Accessed 17 Aug. 2023].

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D., (2020b) Language Models are Few-Shot Learners. [online] Available at: http://arxiv.org/abs/2005.14165.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P. de O., Kaplan, J., Edwards, H., Burda, Y.,
Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G.,
Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F.P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert- Voss, A., Guss, W.H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain,
S., Saunders, W., Hesse, C., Carr, A.N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford,

A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I. and Zaremba, W., (2021a) Evaluating Large Language Models Trained on Code. [online] Available at: http://arxiv.org/abs/2107.03374.

Chen, Z., Chen, W., Smiley, C., Shah, S., Borova, I., Langdon, D., Moussa, R., Beane, M., Huang, T.-H., Routledge, B. and Wang, W.Y., (2021b) FinQA: A Dataset of Numerical Reasoning over Financial Data. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. [online] Stroudsburg, PA, USA: Association for Computational Linguistics, pp.3697-3711. Available at: https://aclanthology.org/2021.emnlp-main.300 [Accessed 8 Aug. 2023].

Clark, K., Luong, M.-T., Le, Q. V. and Manning, C.D., (2020) ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. [online] Available at: http://arxiv.org/abs/2003.10555.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W.,
Yi, J., Zhao, W., Wang, X., Liu, Z., Zheng, H.-T., Chen, J., Liu, Y., Tang, J., Li, J. and Sun, M., (2023) Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence, [online] 53, pp.220–235. Available at: https://www.nature.com/articles/s42256-023-00626-4.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D. and Smith, N.A., (2020) Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. [online] Stroudsburg, PA, USA: Association for Computational Linguistics, pp.8342-8360. Available at: https://www.aclweb.org/anthology/2020.acl-main.740.

Malo, P., Sinha, A., Korhonen, P., Wallenius, J. and Takala, P., (2014) Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, [online] 654, pp.782-796. Available at: https://onlinelibrary.wiley.com/doi/10.1002/asi.23062.

Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E. and Launay, J., (2023) The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. [online] Available at: http://arxiv.org/abs/2306.01116.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., (2019) *Language Models are Unsupervised Multitask Learners*. [online] Available at: https://insightcivic.s3.us-east- 1.amazonaws.com/language-models.pdf [Accessed 17 Aug. 2023].
Shah, A., Vithani, R., Gullapalli, A. and Chava, S., (2023) FiNER: Financial Named Entity Recognition Dataset and Weak-Supervision Model. [online] Available at: http://arxiv.org/abs/2302.11157.

Shah, R., Chawla, K., Eidnani, D., Shah, A., Du, W., Chava, S., Raman, N., Smiley, C., Chen,
J. and Yang, D., (2022) When FLUE Meets FLANG: Benchmarks and Large Pretrained Language Model for Financial Domain. In: *Proceedings of the 2022 Conference on*

Empirical Methods in Natural Language Processing. [online] Stroudsburg, PA, USA: Association for Computational Linguistics, pp.2322–2335. Available at: https://aclanthology.org/2022.emnlp- main.148.

Shao, Z., Gong, Y., Shen, Y., Huang, M., Duan, N. and Chen, W., (2023) Synthetic Prompting: Generating Chain-of-Thought Demonstrations for Large Language Models. [online] Available at: http://arxiv.org/abs/2302.00618.

Song, Q., Liu, A. and Yang, S.Y., (2017) Stock portfolio selection using learning-to-rank algorithms with news sentiment. *Neurocomputing*, 264, pp.20–28.

Sun, J., Zhang, H., Lin, C., Gong, Y., Guo, J. and Duan, N., (2022) APOLLO: An Optimized Training Approach for Long-form Numerical Reasoning. [online] Available at: http://arxiv.org/abs/2212.07249.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E. and Lample, G., (2023) *LLaMA: Open and Efficient Foundation Language Models*. [online] Available at: https://scontent.fmaa12- 2.fna.fbcdn.net/v/t39.8562-6/333078981_693988129081760_4712707815225756708_n.pdf?_nc_cat=108&ccb=1-7&_nc_sid=e280be&_nc_ohc=iVbwJKA_j4MAX9R1vU3&_nc_ht=scontent.fmaa12-2.fna&oh=00_AfB7W0Cxpi6NtifBBNngHfMZkGklmr-mB4tcz8f7jsAYWQ&oe=64DF45E2 [Accessed 15 Aug. 2023].

Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M. and Le, Q. V., (2021) Finetuned Language Models Are Zero-Shot Learners. [online] Available at: http://arxiv.org/abs/2109.01652.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg,D. and Mann, G., (2023) BloombergGPT: A Large Language Model for Finance. [online] Available at: http://arxiv.org/abs/2303.17564.

Yang, Y., UY, M.C.S. and Huang, A., (2020) FinBERT: A Pretrained Language Model for Financial Communications. [online] Available at: http://arxiv.org/abs/2006.08097.

Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong,

Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y. and Wen, J.-R., (2023) A Survey of Large Language Models. [online] Available at: http://arxiv.org/abs/2303.18223.

Wang, Y., Zhao, Y. and Petzold, L., (2023) Are Large Language Models Ready for Healthcare? A Comparative Study on Clinical Language Understanding. [online] Available at: http://arxiv.org/abs/2304.05368.

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., (2017) *Attention Is All You Need*. [online] Available at: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a8 45 aa-Paper.pdf [Accessed 15 Aug. 2023].