

**Financial KOLs' Opinion Mining based on Pre-trained Language Models**

**CS6534 Project 2nd Interim Report**

**Student Name: Wang Yuchen**

**Student Number: 58183945**

**Supervisor: Prof. Song Linqi**

# Introduction

## Background

Individuals known as Key Opinion Leaders (KOLs) hold the power to shape the views and beliefs of their audiences by strategically sharing information, molding news stories, and offering their unique perspectives. As technology continues to evolve, the increasing number of social media users has created a fertile ground for KOLs to emerge and exert their influence on society [1]. In the realm of finance, social media influencers hold substantial sway in bringing together news, offering valuable perspectives, and forecasting market trends through their expressed sentiments. These opinions can rapidly spread and impact investors' approaches and expectations, as financial trading is largely based on anticipation. [2]. This is particularly true for large-scale retail investors who usually have limited knowledge in analysis and rely heavily on expert advice.

It's important to keep in mind that relying solely on Key Opinion Leaders (KOLs) for investment advice may not always be the best course of action. While they may seem trustworthy, there's always the possibility that they could be misleading or even manipulative in their opinions. This can make it difficult for investors to discern the accuracy of their advice. In evaluating the predictive accuracy of authors on Seeking Alpha, who are considered Key Opinion Leaders (KOLs), it has been established that at most 53% of stock performance predictions are accurate, which is only slightly better than random chance [3]. It is therefore deemed risky to blindly follow recommendations from KOLs, as this can lead to undesirable financial losses. In light of this, it is crucial to leverage Natural Language Processing (NLP) techniques to establish a system that analyses the sentiment of such recommendations, evaluates their reliability, and generates valuable investment suggestions. Such a system would effectively assist retail investors in filtering out unreliable information and making informed investment decisions.

Sentiment analysis is a widely studied field that aims to determine the overall sentiment of social media posts. In the financial industry, sentiment analysis is commonly used to predict financial time series. However, the reliability of these predictions is often not justified. Most studies in this area focus on analyzing large, aggregated datasets from popular social media and finance-specific platforms like Twitter, Sina Weibo, Stock Twits, and Seeking Alpha [3]. While these platforms provide valuable insights, they do not take into account the opinions of influential individuals or multimedia platforms like YouTube and Podcast. Considering the impact of multimedia information and the fact that investors do not restrict themselves to text-based resources, it is important to conduct more specific research on selected individuals and incorporate multimedia information in sentiment analysis evaluations.

## Project Objectives & Scope

Based on the existing system, the purpose of this project is to enhance the precision of sentiment analysis in the existing Financial Sentiment Analysis System by adopting advanced models and optimizing the analysis techniques. Specifically, we aim to improve the accuracy of sentiment classification for Key Opinion Leader (KOL) opinions related to stocks. This involves developing more efficient models and fine-tuning strategies to analyze KOLs' opinions and sentiments about various stocks. This project consists of three main research areas:

1. **Optimization models:** Experiment with more pre-trained models such as Llama-2-7B/13B, DeBERTa. These models have been shown to have high accuracy and efficiency in the financial domain. By exploring their feasibility, we can build better financial sentiment analysis models.
2. **Wider use of fine-tuning datasets and training strategies:** In order to enhance the performance of our models, we should consider utilizing a diverse range of Fine-tuning datasets and strategies like LoRa (Low-Rank Adaptation) and observe their impacts on the model's performance. Doing so allows us to explore different possibilities and potentially discover more effective solutions. This will aid us in optimizing the accuracy and efficiency of our models, ultimately leading to better outcomes and improved overall performance.
3. **Evaluation:** Compare the combination of new models and fine-tuning strategies with the existing baseline. Evaluate the advantages and scenarios of the models in terms of performance, efficiency, and adaptability to the task. This will help us to understand which models work best in different situations.

# Related Work

## Sentiment-based Investment Strategy

In the real world, incorporating news and social media sentiment can be beneficial in creating investment strategies. Many studies have investigated the profitability of sentiment-based trading after sentiment analysis implementation.

Karalevicius et al. [4] used a lexicon-based approach to determine the sentiment polarity of Bitcoin-related news. They developed an inter-day sentiment-based trading strategy that goes long if the day's news sentiment is positive and short if it is negative. Following that, in the subsequent two days, the opposite trading action is taken as the market tends to overreact to the news. In the end, the position is reverted again in the hope of adding liquidity to the market and steering the price in a long-term view. This strategy's return and Sharpe ratio outperform the market index and other Bitcoin strategies in most simulation windows. However, it only focuses on Bitcoin and does not include portfolio management, which is a common strategy in investment to diversify unsystematic risk.

de Oliveira Carosia et al. [5] studied the news sentiment of the Brazilian stock market using Artificial Neural Networks. They designed eight sentiment-based investment strategies based on buying a falling stock that is expected to rise and buying a rising stock that is expected to continue increasing. Most of these strategies outperformed Buy & Hold strategies and other benchmarks on both weekly and monthly bases.

Wang et al. [3] conducted sentiment analysis on posts extracted from Seeking Alpha and Stock Twits. The fund was allocated equally among the 500 stocks recommended by the top authors on Seeking Alpha from the previous year, ranked according to the amount of engagement they received. The portfolio undergoes weekly management, which involves selling a stock if the consensus among the top authors regarding it is negative, or holding it (or repurchasing it if previously sold) if the sentiment is positive. This approach has proven to be highly effective, surpassing the S&P 500's Buy & Hold strategy.

The following two pertinent works provide compelling evidence for the profitability of sentiment-based trading strategies on stocks. The methodologies employed in these works may serve as benchmarks and sources of inspiration for the design of trading strategies in this project.

## Language Models Pre-training and Fine-tuning

Pre-training and fine-tuning are two crucial steps in the development and adaptation of Large Language Models (LLMs), such as GPT-3 or BERT. These steps are essential for leveraging the power of these models for specific tasks, such as sentiment analysis, question answering, or named entity recognition.

In natural language processing, large language models (LLMs) are trained on extensive textual data without any specific downstream tasks in mind. This process, known as "pre-training," allows the model to acquire general knowledge from the data it's exposed to, rather than being explicitly taught what to do. Once pre-training is complete, the LLM can be fine-tuned to perform certain tasks. This final step involves teaching the LLM how to perform specific tasks, thereby enabling it to complete more targeted objectives.

Figure 2.1 outlines the process of Pre-Trained Models Followed by Fine-Tuning [6]. The process begins by adding a large amount of data (represented as 1 in the figure) to an untrained model (represented as 2 in the figure), which concludes the "pre-training" phase and results in a Language Model (LLM). The next step involves teaching the LLM a specific task (represented as 3 in the figure). While several tasks could be undertaken, the previous example illustrated the translation of English to French. This step concludes the "fine-tuning" phase, and results in a "fine-tuned" model that can perform the task it was trained to do. In the example cited, this is the point where the individual is capable of generating French translations of English sentences. Subsequently, numbers 4 and 5 demonstrate that once a fine-tuned model is available, an input can be sent into the system, and the fine-tuned model will generate an output.

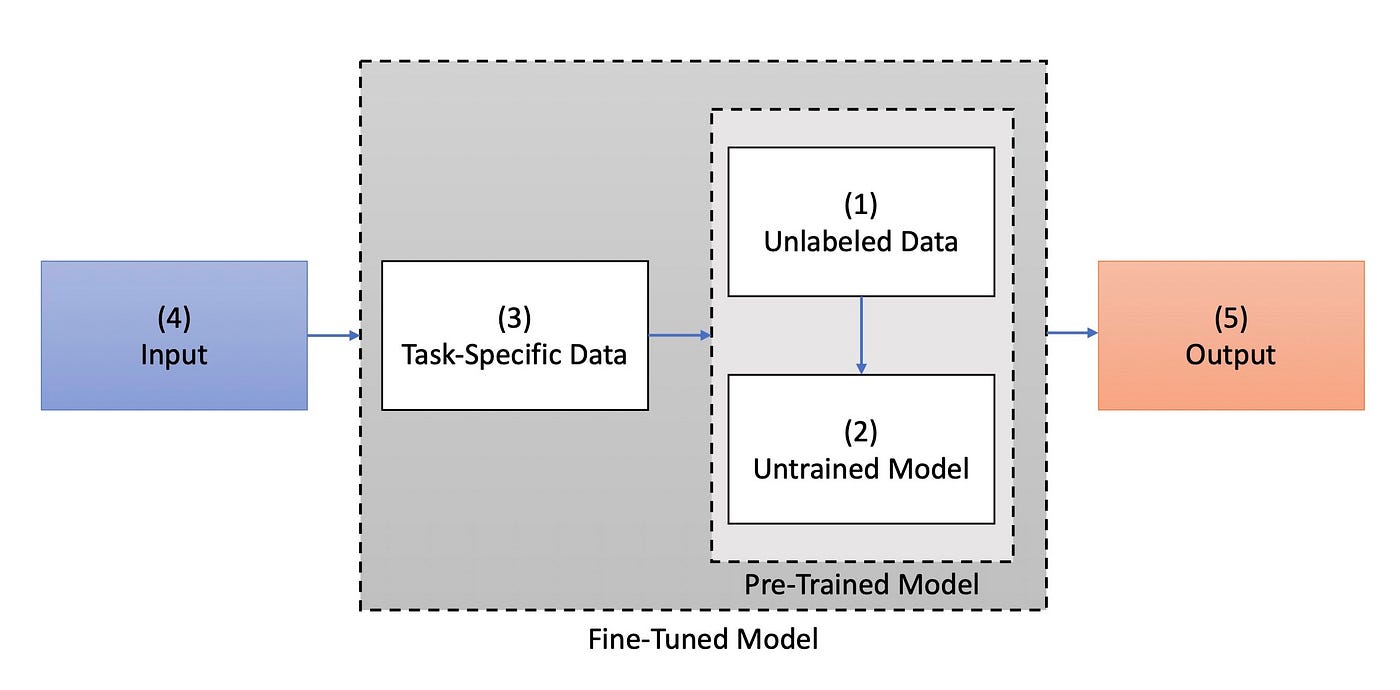


Figure 2.1 General Overview of Pre-Trained Models Followed by Fine-Tuning

# System Modeling

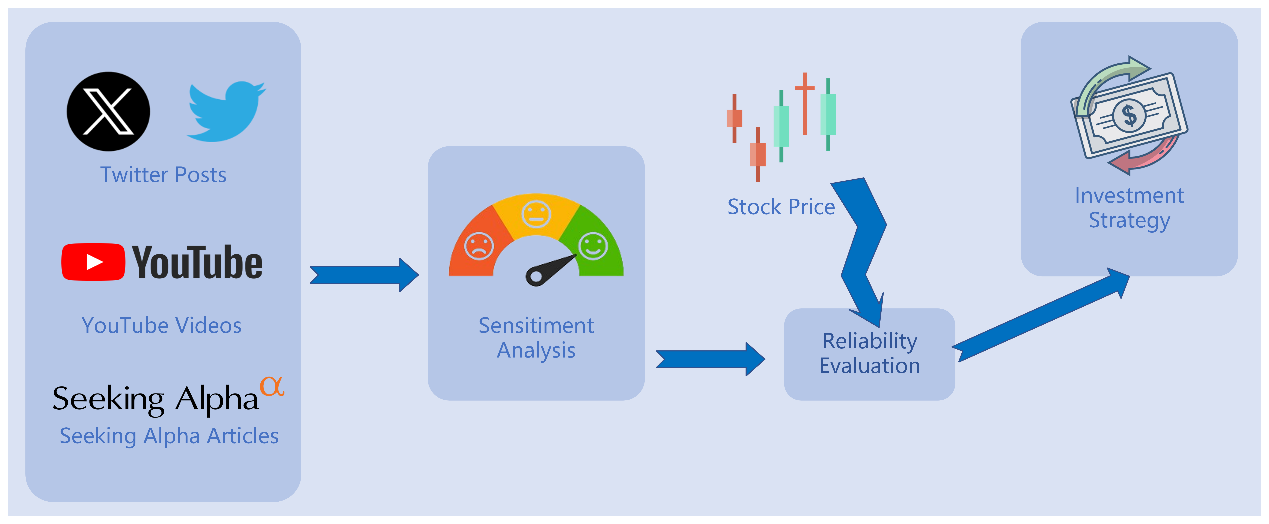


Figure 3.1 Previous System Design Overview

The system previously created is intended to analyze the sentiment of KOL statements, assess their reliability, and formulate profitable investment strategies. Its primary components are depicted in Figure 3.1 to provide a visual representation of the system architecture.

## 3.1 Data Description

This project aims to enhance and optimize the sentiment analysis functionality of the system. To construct data for sentiment analysis, the system uses the existing list of influential financial KOLs on three platforms, namely YouTube, Twitter, and Seeking Alpha, along with their posts on these platforms. After screening and processing the data, the final dataset used for further training and fine-tuning is presented below. We will use the data consistently to train and evaluate improved models, comparing their performance.

1. **Financial Tweets:** There is a significant contrast between the tone and style of tweets versus formal written language, as tweets are often more casual and to the point. As a result, financial tweets are utilized for both further pre-training and fine-tuning. The pre-training stage involves using a Kaggle dataset [7], which comprises 28,275 label-free tweets discussing 453 selected stocks. For fine-tuning, a Hugging Face dataset [8] with 11,932 tweets annotated with three labels (Positive, Neutral, and Negative) is used. This dataset contains a finance-related corpus that is ideal for our purposes.
2. **Financial Phrase Bank:** The dataset [9] is marked as serves as a valuable resource for fine-tuning. It boasts a collection of 4,840 English sentences, sourced randomly from financial news within the LexisNexis database. Each sentence is composed in formal written language and has been thoughtfully annotated by 16 experts in finance and business as either Positive, Neutral, or Negative. The dataset is further segmented into four subsets, based on the level of agreement among annotators: AllAgree, 75Agree, 66Agree, and 50Agree.

## 3.2 Sentiment Analysis

In our existing financial sentiment analysis system, we utilized a pre-trained language model called RoBERTa-Large and fine-tuned it for a classification task. We added a classification head to the pre-trained model which enables it to output the probabilities of each class. We defined three classes that represent the underlying sentiment of the input text, and their meanings are shown in Table 3.1.

|  |  |  |  |
| --- | --- | --- | --- |
| Label | 0 | 1 | 2 |
| Sentiment | Negative | Neutral | Positive |

Table 3.1 Model output illustration

RoBERTa was proposed by Liu et al. [10] as modifications to the BERT [6] pre-training procedure. Compared with BERT, it is trained with dynamic masking, FULL-SENTENCES without the Next Sentence Prediction (NSP) loss, large mini-batches and a larger byte-level BPE. As it removes the next sentence prediction objective, it is more suitable for social media corpus which usually only has one sentence.

RoBERTa-Large is an advanced version of RoBERTa that uses a larger corpus and architecture, resulting in enhanced performance on various fine-tuning tasks. It is a language model that is pre-trained on a significant amount of English data in a self-supervised manner. By replacing the top layer with a task-specific layer, it can be further trained for downstream tasks. The RoBERTa procedure is used to train it over the BERT-Large architecture. Compared to BERT-Base, the BERT-Large architecture has more encoder layers, which boosts the performance of the BERT model on fine-tuning tasks.

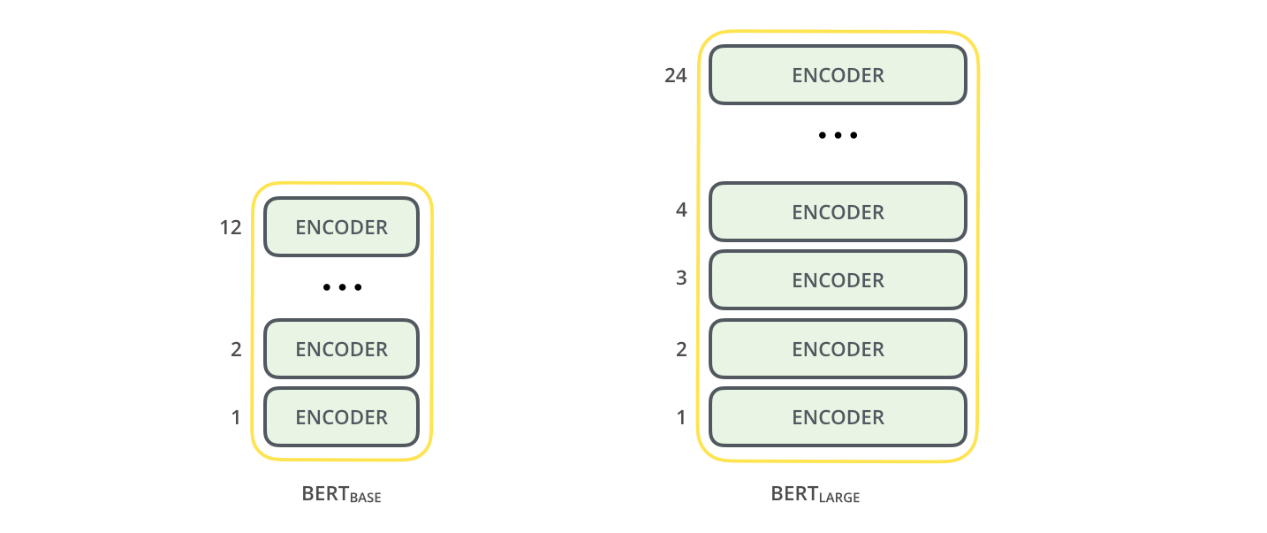


Figure 3.2 Comparison of BERT-Base and BERT-Large architecture

The RoBERTa-Large model has demonstrated superior performance when compared to both the BERT-Large and RoBERTa-Base models across a range of fine-tuning benchmarks, including the General Language Understanding Evaluation (GLUE) benchmark [11]. As a result, the RoBERTa-Large model has been selected for implementation in our existing financial sentiment analysis system.

# Methodology

In the field of Natural Language Processing (NLP), selecting and optimizing the right models is crucial for achieving successful results. Over the past few years, models such as RoBERTa, Llama-2, and DeBERTa have demonstrated strong performance in NLP tasks. For this project, I aimed to explore the Llama-2 and DeBERTa models in-depth by observing their impact on the accuracy of the sentiment analysis system. I will replace the RoBERTa-Large model with these two models to evaluate their performance in the task of sentiment analysis in the financial field.

## Implementation Environment

Python is the go-to language for all stages of our implementations, covering data collection and pre-processing, model construction, KOL reliability evaluation, and building investment strategies. We fine-tune our sentiment analysis model on the PyTorch framework

To train large language models, a powerful graphics card and a high memory system are necessary. In my case, I utilize the High Throughput GPU Cluster server as my experimental environment at first. The CS High Throughput GPU Cluster (HTGC) is a specialized HTCondor cluster with a focus on GPU-related computational applications such as TensorFlow and pytorch. Comprising of one job submission node and 7 job execution slots as Table 4.1, the HTGC is a preferred choice for RoBERTa.

|  |  |
| --- | --- |
| GPU | 7 x Nvidia V100 |
| Maximum memory size per process | 64 GB |
| O/S | Ubuntu 20.04 |
| GPU Memory | 16 GB |
| CUDA Runtime Version | 11.2 |
| CUDA Driver Version | 11.6 |
| CUDA Capability | 7.0 |
| CUDA Device Name | Tesla V100-SXM2-16GB |

Table 4.1 HTGC1 Configuration List

As the experiment progressed, I found it extremely challenging to continue training the llama2 model in the above environment. Training the llama2-7b model on a V100 graphics card that is not capable of acceleration takes more than thirty hours for ten thousand pieces of data, which makes subsequent experimental progress very difficult. With the help of my seniors, I eventually continued the training of llama2-7b on the following environments.

|  |  |
| --- | --- |
| GPU | 4 x Nvidia A40 |
| Maximum memory size per process | 64 GB |
| O/S | Ubuntu 20.04 |
| GPU Memory | 45 GB |
| CUDA Runtime Version | 12.2 |
| CUDA Driver Version | 12.1 |

Table 4.2 Configuration List of Server with A40

## Llama-2

The Llama-2 model is the focus of our research in this project. Llama-2 emerges as a highly advanced language model that surpasses other prominent models like GPT-4/3.5, ChatGPT 4.0, and BERT [12] in terms of performance. Its innovative features and improvements set it apart from the competition. One notable strength of Llama-2 lies in its extensive pre-training data, which has doubled from 1 trillion tokens in Llama-1 to an impressive 2 trillion tokens. This substantial dataset, combined with its ability to process 4096 tokens, enables Llama-2 to generate highly accurate and relevant responses to user queries.

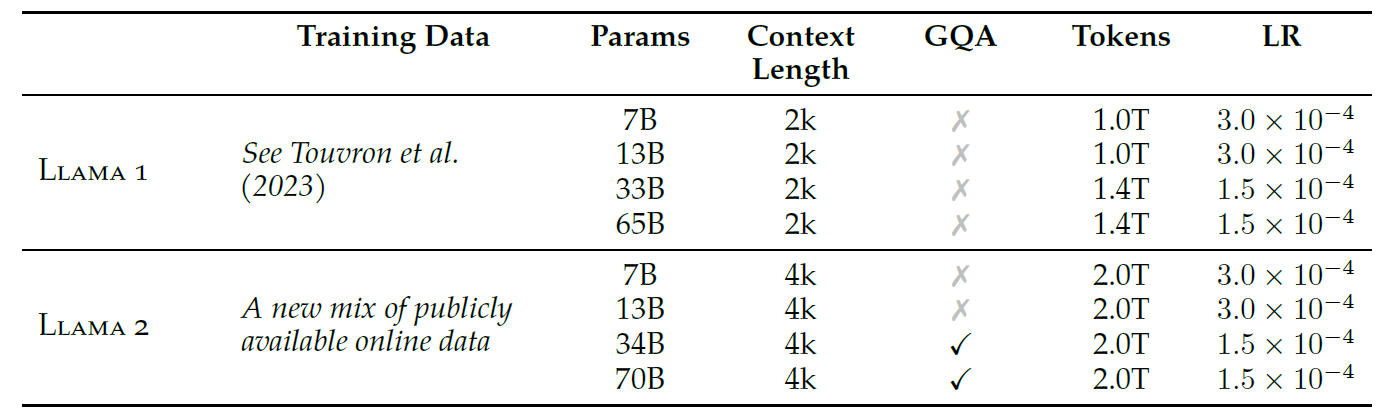


Table 4.3 **Llama 2 family of models.** Token counts refer to pretraining data only. All models are trained with a global batch size of 4M tokens. Bigger models like 34B and 70B use GQA for improved inference scalability.

To achieve its superior performance, Llama-2 incorporates several cutting-edge development paradigms, including rejection sampling, GQA, and GAtt. In detail, the enhancements encompass comprehensive data cleansing processes, refined data combinations, and an expansion of training data by 40%, alongside a doubling of the context length. These modifications leverage grouped-query attention (GQA) to augment inference scalability within larger model frameworks. Comparative analysis presented in Table 4.3 delineates the distinctions between the attributes of the new Llama-2 models and their predecessors, the Llama-1 models. These technological advancements significantly elevate the model's performance across diverse benchmarks.

However, the distinctive feature of Llama-2 lies in its approach to safety and helpfulness. Contrary to other models, Llama-2 employs distinct reinforcement learning from human feedback (RLHF) models tailored specifically for safety and helpfulness. This dual-model strategy ensures that Llama-2 not only delivers pertinent responses but also maintains a high standard of safety by mitigating the risk of generating harmful content. This makes Llama-2 a more reliable and safer option for a broad spectrum of applications.

Furthermore, Llama-2's open-source nature and hardware efficiency make it remarkably versatile and accessible. Its weights can be adjusted according to specific use cases, and it can be fine-tuned on consumer-level hardware. This aspect makes Llama-2 an appealing option for developers and researchers alike. With its extensive training data and the capability for easy customization through prompt-tuning, Llama-2 becomes a powerful tool for various downstream tasks that require specific domain knowledge.

## LoRa

Many applications in natural language processing rely on adopting one large-scale, pre-trained language model that can be applied to multiple downstream applications. This adaptation is typically achieved through fine-tuning, which involves updating all the parameters of the pre-trained model. However, a major drawback of fine-tuning is that the new model contains just as many parameters as the original model. As larger models are trained every few months, this changes from a mere “inconvenience” for GPT-2 (Radford et al., b) or RoBERTa large (Liu et al., 2019) to a critical deployment challenge for GPT-3 (Brown et al., 2020) with 175 billion trainable parameters.

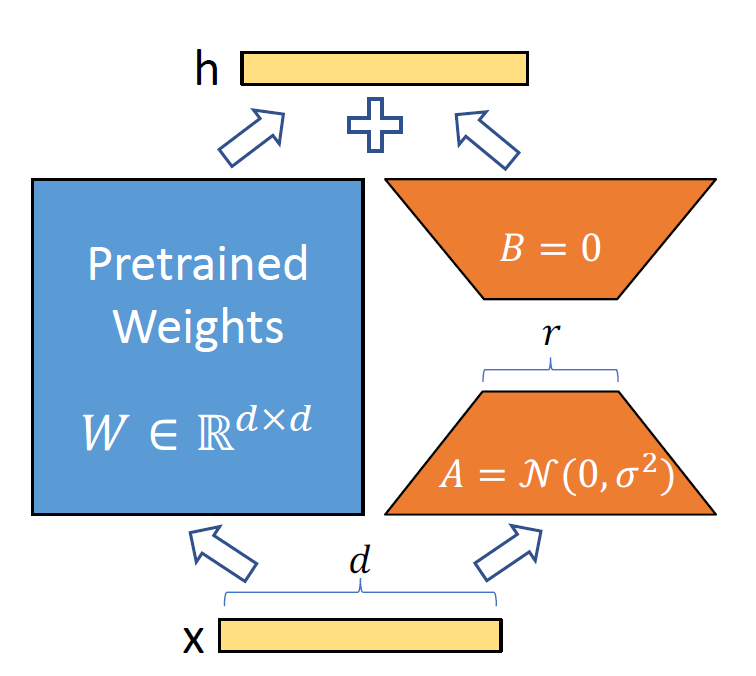


Figure 4.1 Reparameterization of LoRa, only training A and B

Hu, E. J. et al. [16] drew inspiration from the work of Li et al. (2018a) and Aghajanyan et al. (2020) which show that the learned over-parametrized models in fact reside on a low intrinsic dimension. Hu, E. J. et al. hypothesize that the change in weights during model adaptation also has a low “intrinsic rank”, leading to our proposed Low-Rank Adaptation (LoRA) approach. LoRA allows us to train some dense layers in a neural network indirectly by optimizing rank decomposition matrices of the dense layers’ change during adaptation instead while keeping the pre-trained weights frozen, as shown in Figure 4.1. Using GPT-3 175B as an example, we show that a very low rank (i.e., r in Figure 1 can be one or two) suffices even when the full rank (i.e., d) is as high as 12,288, making LoRA both storage- and compute-efficient.

LoRA exhibits several significant advantages that are pivotal for efficient model deployment in various tasks. The architecture allows for the utilization of a pre-trained model across multiple smaller LoRA modules tailored for distinct tasks. By freezing the core model, it facilitates rapid task-switching through the simple replacement of matrices A and B, as depicted in Figure 4.1. This approach substantially diminishes both the storage requirements and the overhead associated with switching tasks.

Furthermore, LoRA enhances training efficiency and significantly reduces the computational demands typically required by hardware. This reduction is achieved by employing adaptive optimizers, which obviate the need for gradient calculations and maintenance of optimizer states for the majority of parameters. Instead, optimization is confined to the smaller, injected low-rank matrices, which are up to three times less demanding in terms of hardware resources. The design of LoRA is inherently linear, which allows for the seamless integration of trainable matrices with the pre-existing frozen weights during deployment. This integration ensures that there is no additional inference latency introduced when compared to a model that is fully fine-tuned.

In the forthcoming experiment, I will apply LoRA to fine-tune the llama2 model. This will demonstrate LoRA's capability to adapt and optimize models efficiently, further reinforcing its suitability for diverse large language model tasks.

# Experiment and Result

## RoBERTa: Further Pre-training

In the construction of the model, we further divide the financial data obtained in the previous section by pre-training and fine-tuning the pre-trained model. A detailed description is presented in Table 5.1. The distribution of each sentiment category in the different datasets is shown in Figures 5.1 and 5.2

|  |  |  |
| --- | --- | --- |
| Dataset | Purpose | Description |
| Unlabeled Financial Tweet | Further pre-training | ▪28275 English financial tweets without label  ▪80% for training and 20% for validation |
| Financial Tweet Sentiment | Fine-tuning | ▪11932 English financial tweets  ▪3 labels (Negative, Neutral & Positive)  ▪80% for training and 20% for testing |
| Financial Phrase Bank | Fine-tuning | ▪English sentences that are selected randomly from financial news in the LexisNexis database.  ▪3 labels (Negative, Neutral & Positive)  ▪The full set (4845 records) and the AllAgree subset (2262 records) are used respectively  ▪80% for training and 20% for testing |

Table 5.1 Dataset description

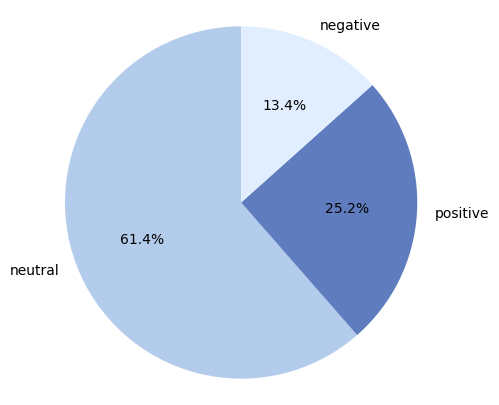


Figure 5.1 Distribution of different sentiment categories in Financial Phrase Bank

To enhance the performance of the sentiment analysis model on financial tweets, we have adopted an advanced pre-training technique. This involves training the RoBERTa-Large model with the Kaggle financial tweets dataset, using the masked language modelling objective.

We use dynamic modelling that generates masking patterns every time a sequence is fed into the model. To enable the model to capture the unique features of financial tweets, we have set the masking probability higher than the original RoBERTa pre-training at 0.15. We have experimented to determine the best masking probability and to prevent overfitting, we have also adopted early-stopping.

## RoBERTa: SMART Framework for Fine-tuning

The SMART framework [14] is a powerful tool for fine-tuning pre-trained language models in a robust and efficient manner. By utilizing this framework, we have been able to achieve new state-of-the-art results on a variety of NLP tasks. To implement it in this project, we have utilized the smart-pytorch Python library [15].

To ensure that the model complexity is effectively controlled during the fine-tuning process, we have incorporated a smoothness-inducing adversarial regularization. This regularization ensures that even if a small perturbation is introduced to the input, the model output will not change significantly. As depicted in Figure 4.10, the decision boundary learned with this regularization (b) is smoother within the neighbourhoods of training data points compared to the one learned without it (a). This valuable property helps to prevent overfitting and improve generalization on a low-resource target domain, such as the financial sentiment analysis task in this project.

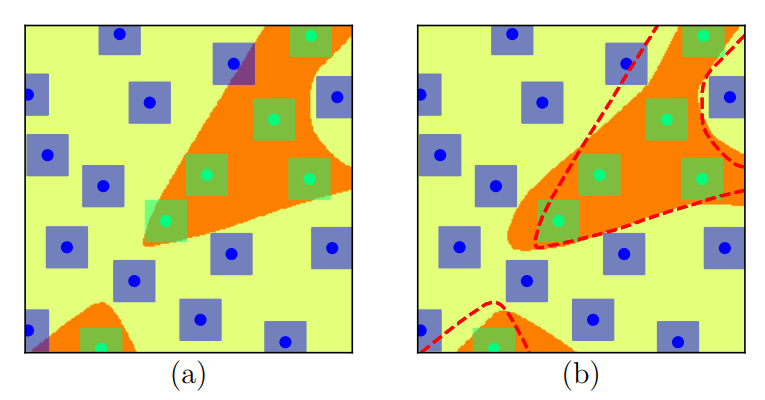


Figure 5.3 The effect of smoothness-inducing adversarial regularization on decision boundary

A set of Bregman proximal point optimization techniques has been suggested to avoid aggressive updating. These methods incorporate a trust-region-like regularization at each iteration to stabilize the fine-tuning process. The updates are then performed within a smaller range of the previous iteration to achieve better results.

* 1. **Llama2-7B: LLM-Finetuning-Toolkit for Fine-tuning**

In this round of experiments, combined with our existing experimental environment, we prioritized the Llama2-7B model with a small number of parameters as our experimental target. To ensure comparability, both models were fine-tuned using identical datasets: short-context financial tweets and long-context entries from the Financial Phrase Bank. The proportion of data allocated to the training and testing sets was meticulously preserved across both models to facilitate an accurate comparative analysis.

During the fine-tuning process, I utilize the LLM Fine-tuning Toolkit, an open-source tool, to fine-tune the Llama2-7B model with our financial data. The LLM Fine-tuning The fine-tuning process was executed using the LLM Fine-tuning Toolkit, a versatile, open-source command-line interface tool designed for streamlined fine-tuning of language models. This toolkit supports a configuration-based approach, enabling researchers to specify experimental parameters through a single YAML configuration file. This file governs various aspects of the experimentation pipeline, including the selection of prompts, the choice of language models, the optimization strategies employed, and the evaluation of model performance. The architecture and functionality of the LLM Fine-tuning Toolkit are depicted in Figure 5.4.

Modifications were made to the prompt.py file within the toolkit to tailor the data loading module for our specific requirements. These adjustments ensured that the financial data was appropriately segmented into training and testing sets in accordance with predefined criteria. A visual representation of the modified code is provided in Figure 5.5, offering insights into the technical adjustments implemented during the fine-tuning process.

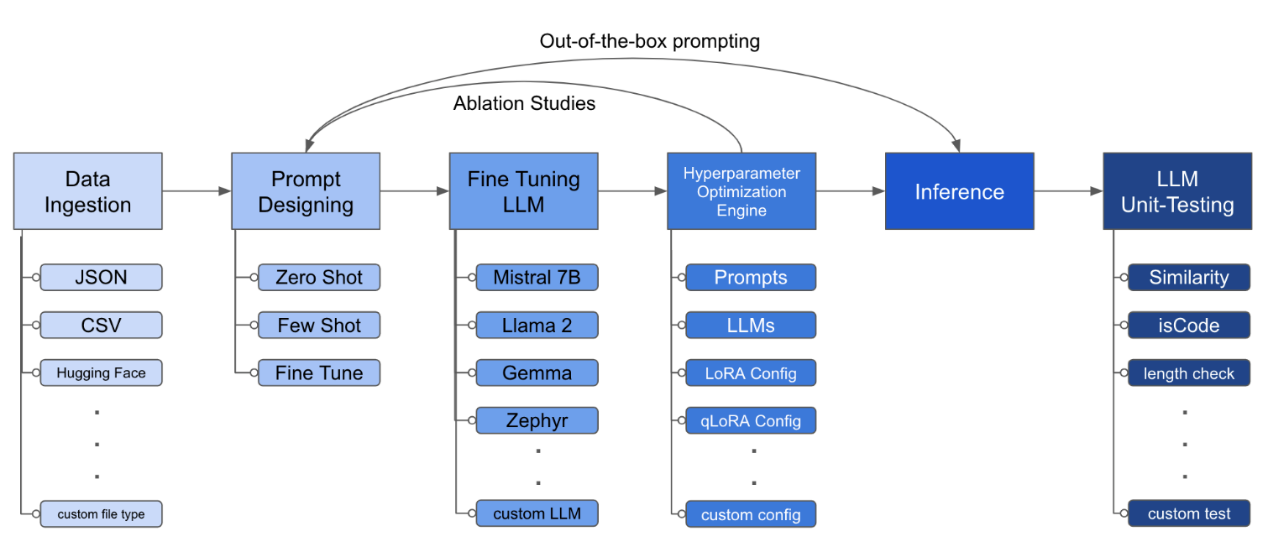


Figure 5.4 The structure of the LLM Fine-tuning Toolkit

****

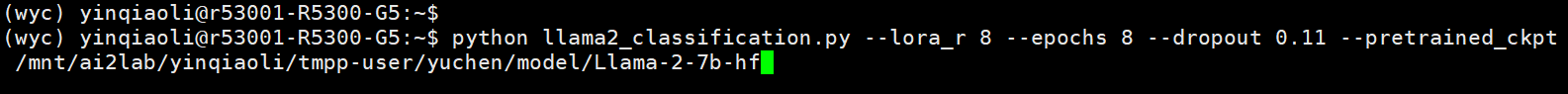
Figure 5.5 Screenshot of prompt.py

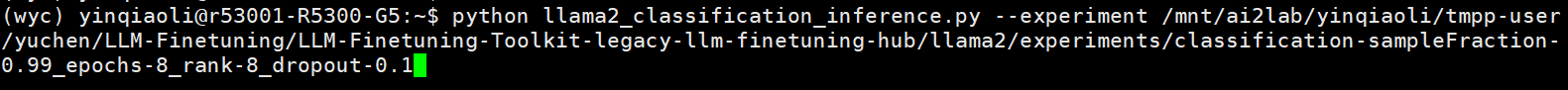
In the llama2\_classfication.py file, we can take control of the training by modifying these hyperparameters. Figure 5.6 shows the screenshot of llama2\_classification.py.

****

Figure 5.6 Screenshot of llama2\_classification.py

Figure 5.7 and Figure 5.8 show how the training of the model and the evaluation of the trained model is carried out via the command line.

****Figure 5.7 Training the llama2 model

****Figure 5.8 Evaluation of the llama2 model

* 1. **RoBERTa: Experiment Result**

This section outlines the process we followed to create an optimal model for sentiment analysis of financial tweets. We conducted further pre-training and fine-tuning, with a specific emphasis on demonstrating the performance of the baseline model for sentiment analysis of short texts related to finance.

In further pre-training, the RoBERTa-Large model is trained again on the Label-free Financial Tweet dataset using the masked language modelling I use the LineByLineTextDataset class from the Hugging Face Transformers library to create training and evaluation datasets based on the unlabeled Financial Tweet dataset. free Financial Tweet dataset. After 41 epochs of training, the loss values change as shown in Figure 5.9.

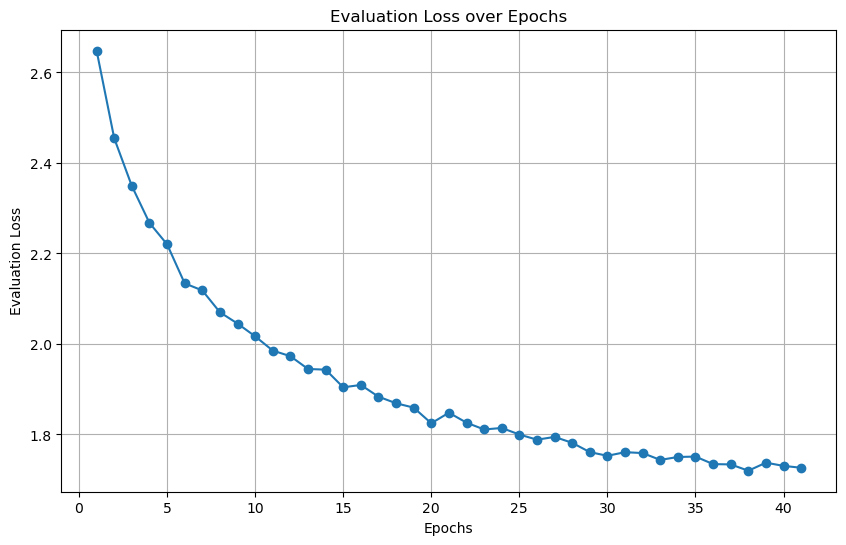


Figure 5.9 Evaluation loss over epochs

I fine-tuned the pre-trained model using labelled financial tweet data from the SMART framework, which was trained using an unlabeled financial corpus. At each epoch, I evaluated the models using the Matthews correlation coefficient (mcc), precision (acc), and F1 score (f1), and the results are presented in Figure 5.10.

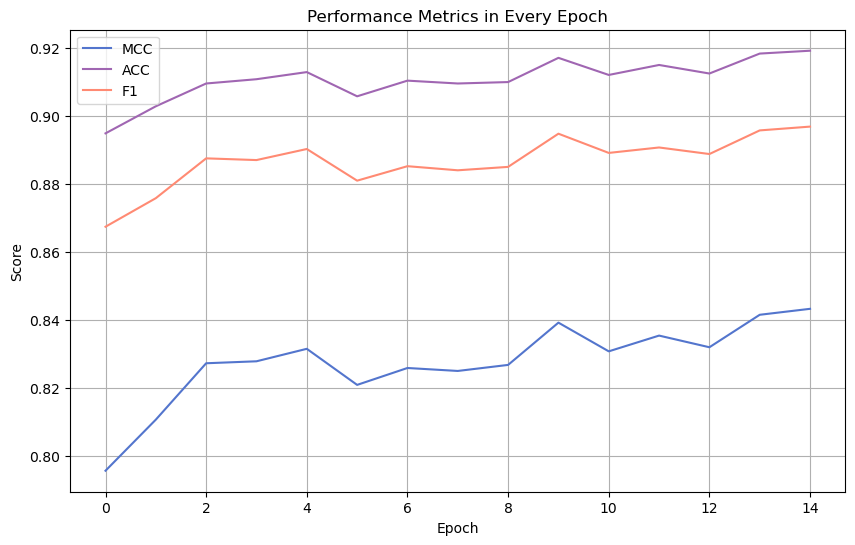


Figure 5.10 Performance Metrics in each epoch in financial tweets

The MCC considers the model's precision, recall, and proportion of true and false-positive cases, providing a more comprehensive assessment, especially in the case of category imbalance. On the other hand, the f1 score reconciles the average of precision and recall and helps measure the model's performance on positive and negative categories. It also takes into account the model's precision and recall, making it particularly useful in the case of category imbalance. In most cases, the closer the values of the mcc and f1 scores are to 1.0, the better the model performs on the classification task.

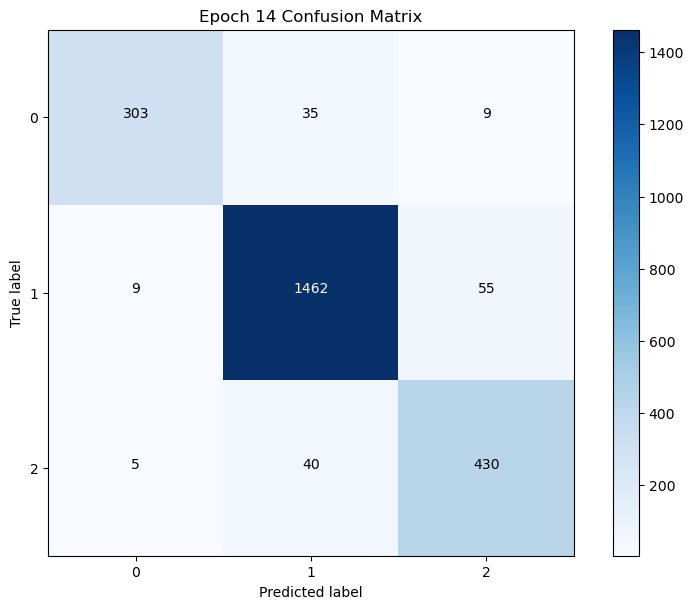


Figure 5.11 Confusion Matrix of RoBERTa-Large-Tweets in Epoch 14

Moreover, Figure 5.11 shows the confusion matrix of the best model obtained in 15 epochs of fine-tuning of RoBERTa-Large in the SMART framework.

Based on further pre-training, we utilized the financial phrase bank to fine-tune the RoBERTa model in order to explore the performance of the RoBERTa model on long text sentiment analysis. I continued to fine-tune the pre-trained model using labelled financial phrase bank data from the SMART framework. At each epoch, I evaluated the models using the Matthews correlation coefficient (mcc), precision (acc), and F1 score (f1), and the results are presented in Figure 5.12. The confusion matrix for the best-performing result is shown in Figure 5.13

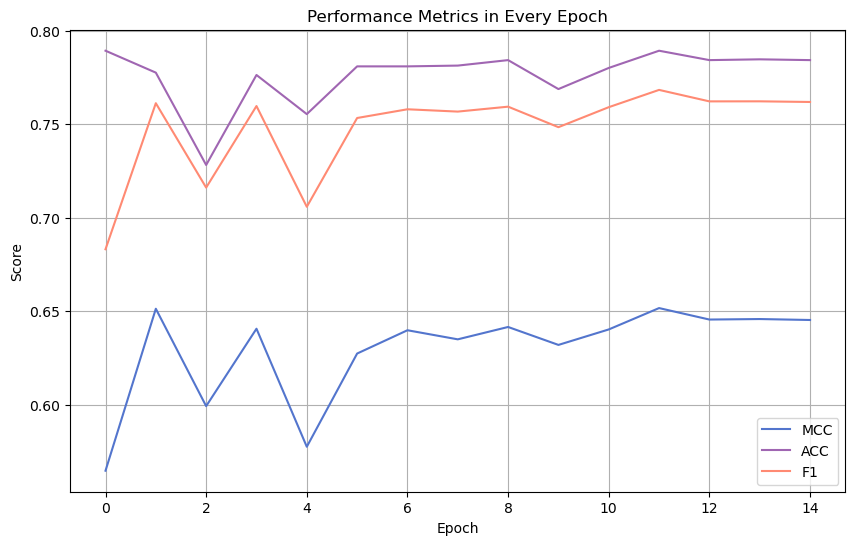


Figure 5.12 Performance Metrics in each epoch in financial phrase bank

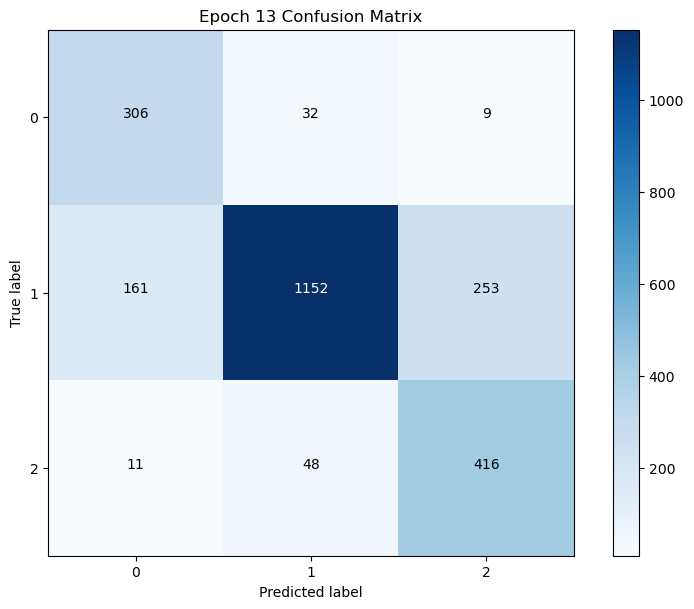


Figure 5.13 Confusion Matrix of RoBERTa-Large-PhraseBank in Epoch 13

* 1. **Llama2-7B: Experiment Result**

The comparative performance analysis of the Llama2-7b model in sentiment analysis tasks for short and long context is systematically presented in Table 5.2. This table juxtaposes the results of the Llama2-7b model with those of the established RoBERTa baseline, facilitating a direct performance comparison.

Additionally, Figures 5.14 and 5.15 illustrate the confusion matrices for the Llama2-7b model, detailing its performance on the test set for short and long context sentiment analyses, respectively. These visual representations provide a clear depiction of the model's classification accuracy and error distribution, offering insightful perspectives into its efficacy across different text lengths.

|  |  |  |
| --- | --- | --- |
|  | F1-Score | Precision |
| RoBERTa-Large-Tweets | 0.895 | 0.915 |
| Llama2-7b-Tweets | 0.822 | 0.815 |
| RoBERTa-Large-PhraseBank | 0.762 | 0.785 |
| Llama2-7b-PhraseBank | 0.938 | 0.930 |

Table 5.2 The performance of Llama2-7b and RoBERTa-large

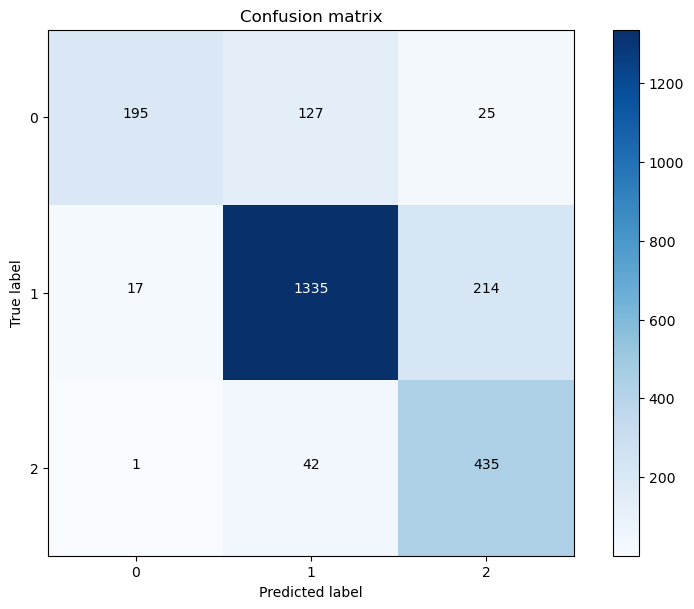
****

Figure 5.14 Confusion Matrix of Llama2-7b-Tweets on test data

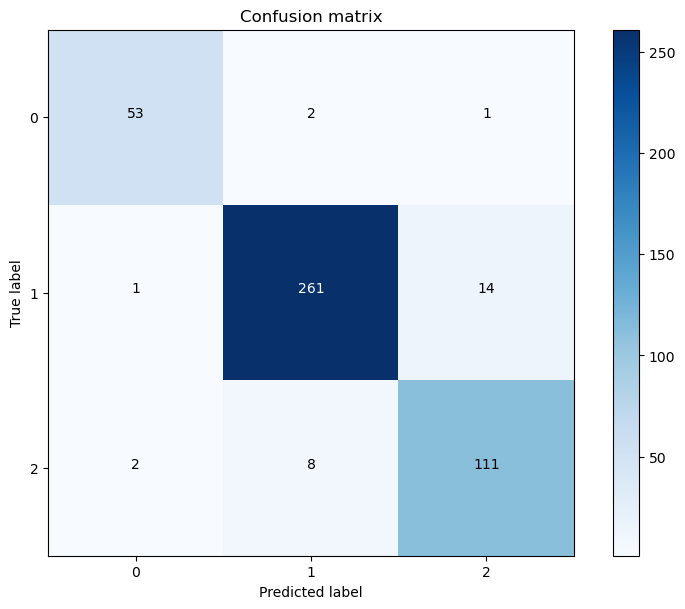
****

Figure 5.15 Confusion Matrix of Llama2-7b-PhraseBank on test data

It is important to highlight that the Llama2-7b model demonstrates superior performance in sentiment analysis of long-context financial news articles, achieving an impressive F-score of 0.938. This contrasts markedly with the performance of the RoBERTa-large model, which exhibits deficiencies in similar tasks. Such a comparison underscores the Llama2-7b model's robust capabilities in processing extensive contextual data.

In the evaluation of Llama2-7b's efficacy in analyzing sentiment within finance-related tweets, the model does not meet the anticipated performance standards nor the established baseline criteria. It is, however, premature to deem the Llama2 model as inherently inadequate for short-text analysis based solely on these findings. Analysis of the confusion matrix indicates a tendency of the model to misclassify negative sentiments as neutral and neutral sentiments as positive. This misclassification suggests the presence of extraneous information in the dataset, which may be adversely affecting the model's training accuracy.

To address these issues, further refinement of the dataset through meticulous data cleaning and preprocessing is essential. Such measures will aim to diminish the influence of data impurities on the training process and potentially enhance the model's performance in short-text sentiment analysis. The refinement of data handling and preprocessing methodologies will constitute a significant focus in the subsequent phase of this research.

# Overall Schedule

## Milestone of the project

|  |  |
| --- | --- |
| Timeline | Task |
| 2024.2.1 ~ until now  (Completed) | Identify and Understand the System Architecture;  Preprocess and Analyze Data;  Establish Baseline Performance with Existing Models; |
| Now ~ 2024.5.24  (Completed) | Understanding the Llama-2 Model;  Integrating Llama-2 into the Existing System;  Evaluating Llama-2 Performance; |
| 2024.5.24 ~ 2024.6.10  (Completed) | Integration and visualization of experimental results;  Further analysis of results;  Finish the second interim report; |
| 2024.6.11 ~ 2024.6.31 | Cleansing of tweets data;  Optimising the performance of the Llama2-7b model on short text financial tweets data; |
| 2024.6.31 ~ 2024.7.15 | Correlating model classification results with stock price forecasts;  Visualization of stock price forecasts; |
| 2024.7.16 ~ 2024.7.28 | Finish both the final report and final presentation. |

## Work to be completed for the next report

In the current phase of our research, I have preliminarily established the viability of the Llama2-7b model by conducting a comparative analysis with the RoBERTa model. This comparison focused on the efficacy of both models in performing sentiment analysis on both short-context and long-context finance-related texts. The findings from this comparative study suggest that the Llama2-7b model holds potential for specialized applications in financial sentiment analysis. Moving forward, the research will concentrate on two primary objectives:

1. **Optimization of the Llama2-7b Model for Short Text Analysis:** The immediate goal is to enhance the performance of the Llama2-7b model specifically for the analysis of short text financial tweets. This optimization process will encompass several dimensions including data preprocessing, fine-tuning of model parameters, and iterative testing to surpass the performance benchmarks set by the RoBERTa model. This step is crucial as it aims to refine the model's accuracy and efficiency in handling real-time, succinct financial data.
2. **Integration into a Financial Sentiment Analysis System:** The second focus of the upcoming research phase involves the integration of the Llama2-7b model into a comprehensive system designed for financial sentiment analysis. This system will utilize the optimized sentiment analysis tool to evaluate statements from influential financial figures (KOLs) and predict stock price movements. The predictive analysis will not only consider historical data trends but also incorporate real-time sentiment shifts to forecast future stock performances. Additionally, the outcomes of these predictions will be meticulously visualized and presented, providing clear and actionable insights into stock market dynamics.

# References

[1] Z. Wang, H. Liu, W. Liu, and S. Wang, “Understanding the power of opinion leaders’ influence on the diffusion process of popular mobile games: Travel Frog on Sina Weibo,” Computers in Human Behavior, vol. 109, p. 106354, Aug. 2020, doi: 10.1016/j.chb.2020.106354.

[2] R. J. Shiller, “Measuring Bubble Expectations and Investor Confidence.” National Bureau of Economic Research, Mar. 1999. doi: 10.3386/w7008.

[3] G. Wang et al., “Crowds on Wall Street: Extracting Value from Social Investing Platforms.” arXiv, Jun. 04, 2014. Accessed: Nov. 09, 2022. [Online]. Available: http://arxiv.org/abs/1406.1137.

[4] V. Karalevicius, N. Degrande, and J. De Weerdt, “Using sentiment analysis to predict interday Bitcoin price movements,” The Journal of Risk Finance, vol. 19, no. 1, pp. 56–75, Jan. 2018, doi: 10.1108/JRF-06-2017-0092.

[5] A. E. de Oliveira Carosia, G. P. Coelho, and A. E. A. da Silva, “Investment strategies applied to the Brazilian stock market: A methodology based on Sentiment Analysis with deep learning,” Expert Systems with Applications, vol. 184, p. 115470, Dec. 2021, doi: 10.1016/j.eswa.2021.115470.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” arXiv, May 24, 2019. Accessed: Nov. 09, 2022. [Online]. Available: <http://arxiv.org/abs/1810.04805>.

[7] N. A, “Zeroshot/twitter-financial-news-sentiment · datasets at hugging face,” zeroshot/twitter-financial-news-sentiment · Datasets at Hugging Face. [Online]. Available: https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment. [Accessed: 30-Mar-2023].

[8] “Financial Tweets | Kaggle.” <https://www.kaggle.com/datasets/davidwallach/> financial-tweets?select=stockerbot-export.csv (accessed Nov. 09, 2022).

[9] P. Malo, A. Sinha, P. Takala, P. Korhonen, and J. Wallenius, “Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts.” arXiv, Jul. 23, 2013. Accessed: Nov. 09, 2022. [Online]. Available: <http://arxiv.org/abs/1307.5336>.

[10] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[11] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, “Glue: A multi-task benchmark and analysis platform for natural language understanding,” Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, 2018.

[12] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

[13] He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654.

[14] H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and T. Zhao, “SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2177–2190. doi: 10.18653/v1/2020.acl-main.197.

[15] Archinetai, “Archinetai/Smart-Pytorch: Pytorch – SMART: Robust and efficient fine-tuning for pre-trained natural language models.,” GitHub. [Online]. Available: https://github.com/archinetai/smart-pytorch. [Accessed: 06-Feb-2023].

[16] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.