# FinGPT: Instruction Tuning Benchmark for Open-Source Large Language Models in Financial Datasets

**Neng Wang**[1*], **Hongyang (Bruce) Yang**[2*], **Christina Dan Wang**[3†]
[1]University of California, Los Angeles; [2]Columbia University;
[3]Shanghai Frontiers Science Center
of Artificial Intelligence and Deep Learning,
NYU Shanghai; Business Division, NYU Shanghai
nengwang19@ucla.edu; hy2500@columbia.edu; christina.wang@nyu.edu

## Abstract

In the swiftly expanding domain of Natural Language Processing (NLP), the potential of GPT-based models for the financial sector is increasingly evident. However, the integration of these models with financial datasets presents challenges, notably in determining their adeptness and relevance. This paper introduces a distinctive approach anchored in the Instruction Tuning paradigm for open-source large language models, specifically adapted for financial contexts. Through this methodology, we capitalize on the interoperability of open-source models, ensuring a seamless and transparent integration. We begin by explaining the Instruction Tuning paradigm, highlighting its effectiveness for immediate integration. The paper presents a benchmarking scheme designed for end-to-end training and testing, employing a cost-effective progression. Firstly, we assess basic competencies and fundamental tasks, such as Named Entity Recognition (NER) and sentiment analysis to enhance specialization. Next, we delve into a comprehensive model, executing multi-task operations by amalgamating all instructional tunings to examine versatility. Finally, we explore the zero-shot capabilities by earmarking unseen tasks and incorporating novel datasets to understand adaptability in uncharted terrains. Such a paradigm fortifies the principles of openness and reproducibility, laying a robust foundation for future investigations in open-source financial large language models (FinLLMs). The codes have been open-sourced at https://github.com/AI4Finance-Foundation/FinGPT.

## 1 Introduction

Natural Language Processing (NLP) stands as a beacon for the financial sector [35, 26, 13], offering groundbreaking opportunities and potential transformations. Large language models (LLMs) [29, 6] grounded in the GPT framework are emerging as a focal point of interest, promising enhanced financial data interpretation and utilization. Instruction Tuning [31, 20] efficiently adapts pre-trained LLMs to specific tasks, significantly saving time and computational resources without starting training from scratch. This cost-effective approach utilizes open-source models through an Instruction Tuning pipeline, achieving or even surpassing the performance of closed-source counterparts with minimal

[*]Equal contribution

[†]Corresponding author: Christina Dan Wang is Assistant Professor, Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning, NYU Shanghai; Business Division, NYU Shanghai, Shanghai China 200122. Email: christina.wang@nyu.edu. Supported in part by National Natural Science Foundation of China (NNSFC) grant 12271363

Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

resource investment. Yet, the challenge remains in adeptly integrating these models, maintaining transparency, and ensuring their seamless adaptability to varied financial tasks.

Existing methods, while revolutionary in their own rights, fall short in certain key areas. Some models face obstacles in effortless integration with diverse financial datasets [16, 17]. Others, although proficient in general contexts, might falter when exposed to intricate financial terminologies and scenarios [2, 33]. Hence, there's an evident need for a more comprehensive, transparent, and adaptable model catering specifically to the financial domain.

In this paper, we propose a novel scheme that utilizes the Instruction Tuning paradigm to magnify the capabilities of LLMs within the financial sphere. The proposed scheme stands out due to its emphasis on transparency, reproducibility, and the plug-and-play nature of model integration. Our works encompass the presentation of the Instruction Tuning paradigm, a deep-dive evaluation into the financial understanding of prevalent models, and an in-depth discussion of our methodological approach, which ensures a consistent training regimen. We bridge the gap between open-source models and financial data, ensuring a harmonious confluence.

Our contributions can be summarized as follows:

- **Instruction tuning paradigm:** We present an Instruction Tuning paradigm, specifically tailored for open-source Large Language Models (LLMs) in the financial sector. This approach not only addresses integration challenges but also enhances the adaptability and relevance of transformer-based models for various financial datasets.
- **Cost-effective benchmarking scheme:** We introduce a benchmarking process designed with a cost-effective and end-to-end training and testing strategy. This scheme is not only tailored for financial contexts but also ensures a comprehensive and systematic evaluation of LLMs from basic competencies to complex, multi-task operations.
- **Deep insights into various base models:** Our work offers a detailed exploration and clarification of various open-source base models such as Llama2, Falcon, ChatGLM2. By highlighting its potential for immediate and transparent integration into the financial sector, we provide valuable insights and plug-and-play guidance for researchers and practitioners working with financial tasks.
- **Promotion of openness and reproducibility:** Our methodology adheres to and advocates for the principles of openness and reproducibility in the research and development of open-source FinLLMs. This contribution lays a solid foundation for future research, facilitating further investigation and development in the field.

The remaining sections of this paper are organized as follows: Section 2 shows the related work; Section 3 delves into the intricacies of the Instruction Tuning paradigm; Section 4 presents our implementation Detail; Section 5 outlines the experiment results; and Section 6 concludes the study with potential future directions.

## 2 Related Works

Recent years have seen a surge in research focused on amalgamating financial datasets with GPT-based models like GPT-3 and GPT-4 [6] for enhanced NLP applications. Generally, there are two prevailing methodologies: Firstly, employing prompt engineering [40, 32, 12] with open-source LLMs, keeping parameters intact; and secondly, using supervised fine-tuning methods such as Instruction Tuning [20] to craft domain-centric LLMs specially designed for financial tasks.

### 2.1 General Large Language Models

- **Llama2** [30] is an open-source LLM developed by Meta, supports 20 languages, building upon its predecessor Llama 1 [29].
- **ChatGLM2** [38, 8] emerges as a bilingual model based on the General Language Model (GLM) framework [9], supporting English and Chinese.
- **BLOOM** [22], is the world's largest open multilingual language model, supports 46 natural languages and 13 programming languages, serving as a comprehensive multilingual solution.

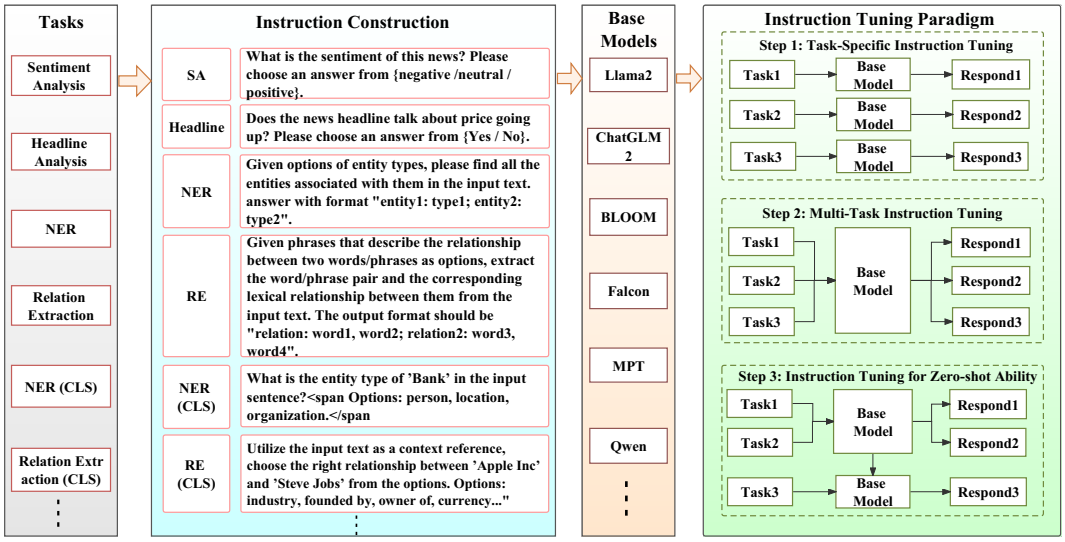


Figure 1: Overview of the proposed Instruction Tuning paradigm

- **Falcon** [1] is renowned for its multilingual support, efficiency in computation, and quality training data from diversified sources.
- **MPT** [27] by MosaicML, pre-trained on English text and code, boasts an optimized architecture, making it efficient for both training and inference tasks.
- **Qwen** [4] from Alibaba stands out for its prowess in both Chinese and English, making it a versatile tool for multilingual applications.

### 2.2 Financial Large Language Models

- **FinBert** [2] is a dedicated model for financial sentiment analysis with under one billion parameters, fine-tuned on a rich financial corpus to excel in finance-specific tasks.
- **FLUE** [23] offers a benchmark derived from five varied financial datasets, acting as an exhaustive evaluation tool for financial language understanding. Its derivative model, FLANG-BERT, outperforms FinBert on these datasets due to domain-specific enhancements.
- **BloombergGPT** [33] is a closed-source model based on BLOOM, trained extensively on diverse financial datasets, thereby encapsulating a broad spectrum of the financial domain.
- **FinGPT** [37, 39, 36] is an open-source LLM, fine-tuned from a general LLM using low-rank adaptation methods [10], fostering accessibility for the broader community.
- **PIXIU** [34] functions as an evaluation benchmark and an instructional dataset. Its focus is solely on the dataset benchmark, exclusively evaluating models derived from Llama without considering other open-source LLMs.

Current research mainly uses Llama models as the base model for financial task evaluations, limiting understanding as different open-source models may excel in various tasks. A broader, more inclusive evaluation encompassing various open-source models could yield insights into task-specific performances and may unveil models that are inherently better aligned with specific financial applications.

Acknowledging these gaps, our work endeavors to present a more sophisticated and integrated paradigm, aiming to seamlessly intertwine open-source LLMs with intricate financial data, thereby proposing a robust solution for domain-specific applications.

## 3 Proposed Paradigm

Initially, we conduct a continuous evaluation of financial tasks while constructing instructions pertinent to each task, followed by the selection and evaluation of a base model. The Instruction Tuning

paradigm outlined in this study, depicted in Figure 1, is carefully designed into three interconnected phases, each playing a crucial role in facilitating the seamless integration and thorough analysis of various financial NLP datasets.

### 3.1 Task-Specific Instruction Tuning

In the initial phase of our paradigm, Task-Specific Instruction Tuning, we meticulously analyze the foundational competencies of LLMs for individual NLP tasks within the finance sector. Each task is examined in isolation during this phase, allowing for a detailed evaluation of LLMs' inherent capabilities and performance. This focused approach generates in-depth insights into the strengths, efficacy, and areas needing improvement for each LLM regarding specific tasks and the ability to efficiently extract, process, and analyze information from various financial data sources.

#### 3.1.1 Potential Challenges in Task-Specific Instruction Tuning

This dedicated approach, while robust, is not without challenges:

- **Varying Task Complexity:** Financial NLP tasks vary considerably in their level of complexity and specificity. As such, LLMs might exhibit proficiency in certain tasks while struggling with others that demand a deeper understanding of the financial domain or more advanced analytical skills.
- **Quality of Financial Datasets:** The quality and reliability of financial datasets used for task-specific instruction tuning directly impact the effectiveness of the tuning process. Ensuring data accuracy, relevance, and completeness while avoiding biased or unrepresentative samples is crucial for the success of this phase.
- **Performance Measurement:** Establishing appropriate metrics and benchmarks to accurately measure and compare the performance of LLMs across various task-specific scenarios can be challenging given the unique characteristics of each task within the financial domain.

#### 3.1.2 Addressing the Challenges in Task-Specific Instruction Tuning

Addressing the outlined challenges requires a multifaceted strategy combining data quality assurance, enhanced model training and testing procedures, and continuous performance monitoring.

1. Recognizing the varied complexity levels across different financial NLP tasks, we implement a dynamic Instruction Tuning approach. This approach is adaptive, changing the depth and breadth of tuning based on the complexity of the task at hand, thus ensuring optimal performance across tasks of varying difficulty and specificity.
2. Rigorous data validation and verification processes are instituted to ensure the quality and reliability of financial datasets utilized in the tuning process. These processes aim to verify data accuracy, completeness, and relevance, thus providing a solid foundation for effective task-specific Instruction Tuning.
3. We devise a set of comprehensive, task-appropriate performance metrics and benchmarks. Continuous refinement and validation of these metrics ensure they remain relevant and reflective of the unique characteristics and requirements of each task.

By addressing these challenges head-on, we ensure that the task-specific phase of our paradigm is both robust and reflective of the unique demands and challenges posed by the financial domain.

### 3.2 Multi-Task Instruction Tuning

The Multi-Task Instruction Tuning phase evaluates the LLMs' versatility and adaptability across concurrent NLP tasks within finance. Here, various instructional tunings are integrated, enabling LLMs to perform multiple tasks simultaneously, offering insight into their multitasking capabilities.

This phase is designed to mirror the complex, multitasking environment that characterizes the financial sector, where the ability to concurrently process and analyze various forms of data is paramount. Therefore, it critically informs the practical utility and efficiency of deploying LLMs in real-world financial scenarios.

#### 3.2.1 Potential Challenges in Multi-Task Instruction Tuning

While this phase is imperative, it introduces several challenges that must be navigatively addressed:

- **Task Interference:** One major challenge is task interference. When models are trained to perform multiple tasks concurrently, the learning for one task might interfere with the learning for another, affecting the overall performance adversely.
- **Computational Complexity:** With the amalgamation of instructional tunings for various tasks, the computational complexity increases. Handling the elevated processing demands without compromising on efficiency and speed becomes challenging.
- **Optimal Task Weighting:** Determining the appropriate balance or weighting among multiple tasks during training to ensure that no single task dominates the learning process is a non-trivial challenge.

#### 3.2.2 Addressing Challenges in Multi-Task Instruction Tuning

Addressing the inherent challenges in Multi-Task Instruction Tuning requires a streamlined approach combining optimized computation, efficient training protocols, and advanced evaluation techniques.

1. Advanced optimization techniques are employed to handle increased computational demands. Through efficient batching and parallel processing, computational loads are effectively distributed, ensuring fast and efficient training without sacrificing model quality.
2. Dynamic task weighting strategies are adopted to facilitate balanced learning across tasks. These adaptive mechanisms adjust the weight assigned to each task's loss during training, promoting harmonious multi-task learning without dominance of any single task.

These strategic measures collectively address the challenges associated with Multi-Task Instruction Tuning, enabling the effective deployment of proficient LLMs in the financial domain.

### 3.3 Instruction Tuning for Zero-shot Ability

The final phase, "Instruction Tuning for Zero-shot Ability," enhances LLMs' zero-shot capabilities. In this crucial phase, LLMs face unprecedented scenarios and tasks, selected to assess their adaptability, learning agility, and response to novel challenges in the financial sector. The introduction of novel datasets and unseen tasks creates a rigorous testing environment for the models. This systematic approach allows for a detailed examination of the LLMs' robustness, flexibility, and problem-solving abilities, essential for operating in the rapidly changing financial landscape.

#### 3.3.1 Potential Challenges in Instruction Tuning for Zero-shot Ability

The complexity and novelty integrated into this phase inevitably introduce an array of challenges, each of which necessitates thoughtful consideration and strategic addressing:

- **Hallucination:** LLMs sometimes generate plausible but unfounded or hallucinated information in their responses, particularly when dealing with unfamiliar or unseen tasks. This phenomenon can lead to misinformation and misinterpretation of the data, which is especially perilous in the financial domain where precision is paramount.
- **Generalization vs. Specialization:** Striking the optimal balance between generalization to new tasks and specialization in previously learned tasks is a perpetual challenge in zero-shot learning environments.

#### 3.3.2 Addressing Challenges in Instruction Tuning for Zero-shot Ability

To effectively navigate through the challenges identified, strategic approaches have been employed:

1. To counteract hallucination issues, we implement a strategy of task reformulation. The models are trained with a more focused objective, making it easier for them to understand and categorize input data without generating extraneous information. This focused training approach narrows down the task gap between the newly reformulated tasks and the target task, thereby fostering a more controlled and accurate generation process.

2. An optimal balance between generalization to novel tasks and specialization in learned tasks is crucial. We use adaptive learning and fine-tuning techniques, dynamically adjusting the learning process based on the task at hand.

Through these carefully devised strategies, we mitigate identified challenges, promoting the effective development and tuning of LLMs for enhanced zero-shot capabilities in the financial domain.

# 4 Implementation Detail

In this section, we expound upon our methodologies in Data Preparation, Instruction Construction, and Training. Comprehensive resources including datasets, codebases, and illustrative examples are accessible at `https://github.com/AI4Finance-Foundation/FinGPT/tree/master/fingpt/FinGPT_Benchmark`.

## 4.1 Data Preparation

In preparing the data, our approach aligns with the methodology utilized by BloombergGPT[33], in which a selection of financial datasets from the FLUE benchmark[23] is adopted for various tasks.

**Selection of Datasets:** For the Sentiment Analysis (SA) task, we leverage datasets FPB[17] and FiQA-SA[16], while for the Headline Classification (HC) task, the Headline dataset[25] is employed. Furthermore, the NER dataset[21] is utilized for Named Entity Recognition (NER) tasks. To enhance diversity within the data, particularly for the sentiment analysis task, additional datasets, TFNS[15] and NWGI[36], were incorporated into the mix. This step was crucial to ensure that the models developed had exposure to a wide variety of data, promoting robustness and versatility in their application.

**Financial Relation Extraction:** Additionally, our implementation ventured into the domain of financial Relation Extraction (RE). For this endeavor, we engaged with the FinRED dataset[24], providing a basis for the extraction and understanding of financial relations within the textual data.

**Rationale Behind Dataset Choices:** Datasets were carefully selected to cover a wide range of financial NLP tasks, with each contributing uniquely to the model's understanding and performance on specific tasks. Care was taken to ensure that the datasets were complementary, and their integration would facilitate a comprehensive and nuanced understanding of the model's capabilities and areas that required further refinement and tuning.

| Task | Dataset | Total Samples |
|---|---|---|
| Sentiment Analysis (CLS) | FPB | 3634 |
| | FiQA-SA | 938 |
| | TFNS | 9543 |
| | NWGI | 16184 |
| Named Entity Recognition | NER | 609 |
| Headline Classification (CLS) | Headline | $11412 \times 9$ |
| Relation Extraction | FinRED | 6768 |
| Named Entity Recognition (CLS) | NER | 1003 |
| Relation Extraction (CLS) | FinRED | 9657 |

Table 1: Overview of tasks and datasets: The trailing (CLS) means the task can be formatted into a unified classification task. Number of samples are counted before augmented by hand-craft/gpt-generated prompts. Headline contains 11412 sample and 9 question for each sample. For each entity in NER, we ask models to classify its entity-type forming NER(CLS). For each relation in RE, we ask models to classify its relation-type forming RE(CLS).

### 4.2 Instruction Construction

Constructing precise instructions is pivotal for both task-specific and multi-task Instruction Tuning, with each task being guided by a unique instruction prompt.

**Template Structure:** The instruction template is structured as follows:

**Instruction**: [prompt] **Input**: [input] **Answer**: [output]

This template provides a standardized format, facilitating consistency across different tasks and experimental setups.

**Instruction Formulation for Specific Tasks:**

- **Sentiment Analysis (SA) Task:** Instructions for the SA task are directly adopted from FinGPT [39], leveraging their previously established efficacy.
- **Headline Classification (HC) Task:** For the HC task, we have chosen to use Bloomberg's [33] set of instructions, taking advantage of their industry-aligned approach.
- **Named Entity Recognition (NER) and Relation Extraction (RE) Tasks:** In the cases of NER and RE, instructions were meticulously crafted in-house to address the specific nuances and requirements of these tasks.

**Multi-Task and Zero-Shot Experiment Adjustments:** To facilitate multi-task and zero-shot experiments, modifications were made to the NER and RE tasks, reformatting them into classification tasks—dubbed NER(CLS) and RE(CLS). This alignment with SA and HC tasks not only enriched our set of tasks but also aimed to enhance the generalization capabilities of the LLMs.

**Zero-Shot Experiment Instructions:** An [Options] section is added to each instruction for standardization in zero-shot experiments, streamlining instructions and limiting unexpected answers. ChatGPT was used to create ten unique instructions per task, with option order randomized in the training set for diversity. The updated zero-shot experiment template is provided below:

**Instruction**: [prompt] **Options**: [options] **Input**: [input] **Answer**: [output]

Concrete examples illustrating the constructed instructions are depicted in Figure 1, providing visual insights into the practical application of the instruction set within the experimental framework.

### 4.3 Training Detail

In our research, six open-source LLMs—Llama2-7B[18], Falcon-7B[28], BLOOM-7.1B[5], MPT-7B[19], ChatGLM2-6B[3], and Qwen-7B[7]—are selected as the subjects for the application of our Instruction Tuning paradigm. Each of these models is of a comparable size and is used in their base form, without the inclusion of any instruction-tuned variants or chat versions, with the singular exception of ChatGLM2 (the base model of which has not been released).

**Utilizing LoRA.** Given the substantial computational resources necessitated by the fine-tuning process of these LLMs, our approach incorporates the use of LoRA[10], maintained with a rank of 8 and a scaling factor (alpha) of 32, targeting the projection layers within attention modules.

In the three phases of Instruction Tuning:

- **Task-specific job**: epochs are set to 8 for SA, HC, and RE tasks due to their ample sample sizes. The NER task, possessing a limited sample size, warrants the setting of epochs to 50.
- **Multi-task job**: the models are exposed to a combined dataset of SA, HC, NER, RE, NER(CLS), and RE(CLS) for a total of 4 epochs. Tasks characterized by smaller sample sizes are oversampled to maintain balance.
- **Zero-shot job**: the models undergo fine-tuning on a consolidated dataset comprised of NER(CLS), RE(CLS), and HC for a single epoch, subsequently undergoing evaluation on the SA task. Checkpoints are systematically saved every 100 steps and are selected based on their evaluation loss pertaining to the SA task.

**Model Parameters:** Experiments used four RTX 3090 GPUs, with max token length of 512 and per-device batch size of four, plus eight gradient accumulation steps. We employed AdamW optimizer[14],

with an initial learning rate of $1 \times 10^{-4}$, linearly decaying to zero following a 3% warm-up in steps, utilizing FP16 precision for cost-effectiveness.

**Training Cost Analysis:** With GPU hourly rate at $3.36, task-specific jobs for six base models across four tasks took 30 hours. Multi-task and zero-shot jobs required about 60 hours due to increased instructions per base model, totaling 90 hours. Thus, the entire training cost was $302.4.

| Dataset | Llama2 | Falcon | MPT | BLOOM | ChatGLM2 | Qwen |
|---|---|---|---|---|---|---|
| SA | 0.820 (2) | 0.804 (4) | **0.821** (1) | 0.748 (6) | 0.798 (5) | 0.811 (3) |
| NER | 0.673 (3) | 0.619 (5) | 0.615 (6) | **0.729** (1) | 0.645 (4) | 0.679 (2) |
| HC | **0.942** (1) | 0.940 (3) | 0.938 (4) | 0.930 (6) | **0.942** (1) | 0.936 (5) |
| RE | 0.395 (3) | **0.428** (1) | 0.309 (6) | 0.425 (2) | 0.340 (5) | 0.371 (4) |
| Avg Ranking | **2.0** | 3.25 | 4.25 | 3.75 | 3.75 | 3.5 |

Table 2: Task-Specific Instruction Tuning Results Summary: Each row presents the F1-score of base models tuned on the specified task, along with their rankings in the bracket. The best model in each task is highlighted in bold. The average of rankings is computed to assess overall performance.

# 5 Experiment Results

## 5.1 Task-Specific Instruction Tuning

**Result Overview.** Table 2 summarizes the results of task-specific Instruction Tuning. For Sentiment Analysis (SA), we document the average performance of models across all sub-datasets, represented through the mean F1-score, with detailed performance metrics for each SA dataset available in Table 3. The entity-level F1-score is reported for Named Entity Recognition (NER), whereas for Health Classification (HC) and Relation Extraction (RE), the F1-scores are reported respectively. For RE, we solely consider the F1-score for identified relations, excluding the (relation, subject, object) tuples.

**Performance Insights.** The experimental findings yield interesting insights. Notably, Llama2 delivers superior overall performance, as evidenced by its average ranking of second place across all tasks. Both Falcon and Qwen demonstrate versatility, yielding balanced performances across all tasks under consideration. In contrast, while BLOOM excels significantly at Information Extraction (IE) tasks such as NER and RE, it falls short in classification tasks like SA and HC, where it records the lowest performance. Although MPT secures the highest score in SA, it underperforms in NER and RE tasks.

| Phase | Dataset | Llama2 | Falcon | MPT | BLOOM | ChatGLM2 | Qwen |
|---|---|---|---|---|---|---|---|
| Task-Specific | FPB | 0.863 | 0.846 | **0.872** | 0.810 | 0.850 | 0.854 |
| | FiQA | **0.871** | 0.840 | 0.863 | 0.771 | 0.864 | 0.867 |
| | TFNS | 0.896 | 0.893 | **0.907** | 0.840 | 0.859 | 0.883 |
| | NWGI | **0.649** | 0.636 | 0.640 | 0.573 | 0.619 | 0.638 |
| | Avg | 0.820 | 0.804 | **0.821** | 0.748 | 0.798 | 0.811 |
| Multi-Task | FPB | 0.861↓ | 0.845↓ | 0.870↓ | 0.766↓ | 0.836↓ | **0.873**↑ |
| | FiQA | 0.825↓ | **0.881**↑ | 0.863- | 0.737↓ | 0.822↓ | 0.870↑ |
| | TFNS | 0.890↓ | 0.880↓ | **0.892**↓ | 0.789↓ | 0.858↓ | 0.890↑ |
| | NWGI | 0.652↑ | 0.647↑ | 0.651↑ | 0.530↓ | 0.618↓ | **0.653**↑ |
| | Avg | 0.807 | 0.813 | 0.819 | 0.701 | 0.784 | **0.822** |
| Performance Gain | | -1.3% | +0.7% | -0.2% | -4.7% | -1.4% | **+1.1%** |

Table 3: Sentiment Analysis Instruction Tuning Results: The table reports detailed F1-scores for base models tuned during task-specific and multi-task phases on each sentiment analysis dataset. Arrows (↑↓) denote the influence of multi-task settings on Instruction Tuning results, with performance gains calculated between phases based on average F1 scores across all datasets.

| Task | Phase | Llama2 | Falcon | MPT | BLOOM | ChatGLM2 | Qwen |
|---|---|---|---|---|---|---|---|
| NER | Task-Specific | 0.637 | 0.619 | 0.615 | **0.729** | 0.645 | 0.679 |
| | Multi-Task | 0.678↑ | 0.600↓ | 0.682↑ | **0.709**↓ | 0.629↓ | 0.666↓ |
| | Performance Gain | +4.1% | -1.9% | **+6.7%** | -2.0% | -1.6% | -1.3% |
| HC | Task-Specific | **0.942** | 0.940 | 0.938 | 0.930 | **0.942** | 0.936 |
| | Multi-Task | **0.938**↓ | 0.932↓ | 0.928↓ | 0.898↓ | 0.932↓ | 0.922↓ |
| | Performance Gain | **-0.4%** | -0.8% | -1.0% | -3.2% | -1.0% | -1.4% |
| RE | Task-Specific | 0.395 | **0.428** | 0.309 | 0.425 | 0.340 | 0.371 |
| | Multi-Task | 0.674↑ | 0.576↑ | 0.667↑ | **0.697**↑ | 0.557↑ | 0.640↑ |
| | Performance Gain | +27.2% | +14.8% | **+35.8%** | +27.2% | +21.7% | 26.9% |

Table 4: Multi-Task Instruction Tuning Summary: The table reports entity-level F1 scores for NER, relation-only F1 for RE, and standard classification F1 for HC. It includes both task-specific and multi-task models for comparison. Arrows (↑↓) signify performance gains from multi-task settings, calculated in each task's last row.

### 5.2 Multi-Task Instruction Tuning

**Performance Evaluation Setup.** This subsection presents the performance evaluation of various LLMs following multi-task Instruction Tuning, and it provides a comparative analysis between models that underwent multi-task and task-specific tuning. Each model was trained on an aggregated dataset encompassing SA, HC, NER, RE, NER(CLS), and RE(CLS). Table 3 displays the results on the SA task and its sub-datasets, while Table 4 shows the results for NER, HC, and RE tasks.

**Performance in Classification Tasks.** In classification tasks, such as SA and HC, most models displayed a minor decline in performance when concurrently trained with unrelated tasks. An exception to this trend, Qwen consistently enhanced its performance across all SA sub-datasets. Additionally, Falcon also showed improvement in specific areas. Conversely, BLOOM, which already had limited success during the task-specific tuning phase, suffered the most substantial performance degradation in classification tasks after multi-task tuning.

**Performance in Information Extraction Tasks.** The scenario differs for information extraction tasks like NER and RE. For RE, all models exhibited significant improvement, likely due to the incorporation of RE(CLS) and additional financial NLP tasks, suggesting potential under-fitting during the task-specific tuning phase. While Llama2 and MPT exhibited progress in NER, not all models mirrored this improvement. Notably, these two models also displayed the most substantial enhancements in RE. BLOOM, in particular, made significant strides, outperforming Falcon and reaffirming its dominance in information extraction tasks.

**Model Improvement Analysis.** Despite its unsatisfying performance in the task-specific phase, MPT registered the most significant improvement in both NER and RE tasks, while Falcon and ChatGLM2 experienced moderate gains in both tasks, further underlining the interconnected performance dynamics in similar tasks.

**Summary of Findings.** Llama2 and MPT not only displayed versatility by excelling in various tasks but also benefited from the multi-task learning environment. While models like Falcon, ChatGLM2, and Qwen maintained their baseline performances, BLOOM managed to enhance its strengths minimally. However, its weaknesses became more pronounced in a multi-task setting.

| Dataset | Llama2 | Falcon | MPT | BLOOM | ChatGLM2 | Qwen |
|---|---|---|---|---|---|---|
| FPB | 0.621 | **0.791** | 0.599 | 0.576 | **0.803** | 0.576 |
| FiQA | 0.565 | **0.625** | 0.591 | 0.517 | **0.631** | 0.517 |

Table 5: Zero-shot Sentiment Analysis Results: The zero-shot F1-scores on the Sentiment Analysis test datasets are reported for all base models. The two best models are highlighted in bold.

### 5.3 Instruction Tuning for Zero-shot Ability

**Training Setup and Modification.** This subsection delineates the results of our examination on the zero-shot abilities of LLMs post-Instruction Tuning. For this evaluation, the LLMs were trained on three distinct classification tasks: HC, NER(CLS), and RE(CLS). Initially, we endeavored to employ the original NER and RE tasks for training purposes. However, these did not sufficiently activate the models' zero-shot abilities due to insufficient instruction diversity. This limitation led to issues such as model hallucination and inconsistency in response to the provided options.

**Task Reformulation to Mitigate Hallucination.** To mitigate this, we reformulated NER and RE tasks into classification tasks, thereby narrowing the task gap with the target SA task. Furthermore, considering the challenge in distinguishing between neutral sentiments and positive/negative ones in a zero-shot setting, all samples labeled "neutral" were excluded from the study.

**Comparative Performance Insights.** Table 5 presents the mean F1-score results for FiQA and FPB. ChatGLM2 stood out in zero-shot tasks, likely due to its chat-centric tuning. Falcon followed closely in overall performance. Llama2 and MPT, while lagging behind ChatGLM2 and Falcon, showed promise in classifying a subset of the samples. However, both BLOOM and Qwen struggled, often misclassifying responses as "positive". BLOOM's results were expected given its past struggles with classification, but Qwen's suboptimal performance was surprising.

**Generalization Capability Insights.** These findings are particularly illuminating as neither ChatGLM2 nor Falcon excelled in the task-specific and multi-task Instruction Tuning phases but demonstrated significant generalization capabilities, understanding unseen instructions, and making accurate classifications during the zero-shot tasks.

## 6 Conclusion and Future Work

In conclusion, this paper presented an Instruction Tuning paradigm that includes task-specific, multi-task, and zero-shot instruction tuning of LLMs within the financial sector. Our work articulated and showcased the diverse capabilities and potential limitations of different LLMs when subjected to various NLP tasks integral to finance. Through rigorous experimentation and analysis, the paper unveiled distinct performance patterns and offered valuable insights into how these models can be efficiently and effectively employed for specific financial applications.

Future work will focus on integrating additional open-source base models, investigating larger models with parameter sizes between 13 and 100 billion, and deepening efforts to enhance the robustness and generalization capabilities of Large Language Models (LLMs). We'll also develop strategies to reduce task interference and hallucination [11], ensuring accurate and reliable model responses across diverse tasks.

## References


[1] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. Falcon-40B: an open large language model with state-of-the-art performance. 2023.

[2] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[3] Knowledge Engineering Group (KEG) & Data Mining at Tsinghua University (THUDM). Chatglm2-6b. *https://huggingface.co/THUDM/chatglm2-6b*, 2023.

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[5] BigScience. Bloom-7.1b. *https://huggingface.co/bigscience/bloomz-7b1*, 2023.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[7] Alibaba Cloud. Qwen-7b. *https://huggingface.co/Qwen/Qwen-7B*, 2023.

[8] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022.

[9] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: general language model pretraining with autoregressive blank infilling. pages 320–335, 2022.

[10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2021.

[11] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[12] Vivian Liu and Lydia B Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–23, 2022.

[13] Xiao-Yang Liu, Ziyi Xia, Jingyang Rui, Jiechao Gao, Hongyang Yang, Ming Zhu, Christina Wang, Zhaoran Wang, and Jian Guo. FinRL-Meta: Market environments and benchmarks for data-driven financial reinforcement learning. *Advances in Neural Information Processing Systems*, 35:1835–1849, 2022.

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[15] Neural Magic. Twitter financial news sentiment. *https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment*, 2022.

[16] Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. WWW'18 open challenge: financial opinion mining and question answering. In *Companion Proceedings of the the Web Conference*, pages 1941–1942, 2018.

[17] Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796, 2014.

[18] Meta. Llama-2-7b. *https://huggingface.co/meta-llama/Llama-2-7b*, 2023.

[19] Mosaicml. Mpt-7b. *https://huggingface.co/mosaicml/mpt-7b*, 2023.

[20] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[21] Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, Parramatta, Australia, December 2015.

[22] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[23] Raj Sanjay Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. When flue meets flang: Benchmarks and large pre-trained language model for financial domain. *arXiv preprint arXiv:2211.00083*, 2022.

[24] Soumya Sharma, Tapas Nayak, Arusarka Bose, Ajay Kumar Meena, Koustuv Dasgupta, Niloy Ganguly, and Pawan Goyal. Finred: A dataset for relation extraction in financial domain. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 595–597, New York, NY, USA, 2022. Association for Computing Machinery.

[25] Ankur Sinha and Tanmay Khandait. Impact of news on the commodity market: Dataset and results, 2020.

[26] Yen-Jen Tai and Hung-Yu Kao. Automatic domain-specific sentiment lexicon generation with label propagation. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, pages 53–62, 2013.

[27] MosaicML NLP Team. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. Accessed: 2023-05-05.

[28] Technology Innovation Institute (TII). Falcon-7b. *https://huggingface.co/tiiuae/falcon-7b*, 2023.

[29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[30] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[31] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[32] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*, 2023.

[33] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. BloombergGPT: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

[34] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *arXiv preprint arXiv:2306.05443*, 2023.

[35] Frank Z Xing, Erik Cambria, and Roy E Welsch. Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, 50(1):49–73, 2018.

[36] Hongyang Yang. Data-centric fingpt. open-source for open finance. `https://github.com/AI4Finance-Foundation/FinGPT`, 2023.

[37] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023.

[38] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

[39] Boyu Zhang, Hongyang Yang, and Xiao-Yang Liu. Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models. *arXiv preprint arXiv:2306.12659*, 2023.

[40] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022.