Benchmarking LLM Performance on Financial Tasks: Zero is Not Hero Yet Replicated and Expanded

Mika Okamoto

Abstract

Large language models (LLMs) have been an increasingly popular topic of conversation and research in recent times, due to their impressive performance on a wide variety of tasks with little additional training. In this paper, I focus on the effectiveness of LLMs in the financial domain. I aim to replicate the work in comparing zero-shot LLM performance against fine-tuned RoBERTa in the finance domain from Shah and Chava (2023), and then expand upon it through few-shot LLM prompting and the newly popular open-source LLM framework DSPy. Our findings demonstrate that ChatGPT performs well without labeled data, and that adding labeled data with few-shot prompting decreases ChatGPT's performance, but fine-tuned RoBERTa outperforms these LLMs. However, algorithmically optimizing prompts with DSPy shows promise and higher performance in the LLM field. Our codebase is publicly available on GitHub[1].

1 Introduction

OpenAI's ChatGPT[2] revolutionized the natural language processing (NLP) field with its shockingly high performance on various NLP tasks with little to no fine tuning, as seen by Qin et al. (2023). Thus, I seek to determine exactly how good ChatGPT's capabilities are compared to the current standard of fine-tuned PLMs, and if they can be improved with other prompting strategies.

This paper focuses on replicating and expanding upon the work done in Zero is Not Hero Yet: Benchmarking Zero-Shot Performance of LLMs for Financial Tasks (Shah and Chava, 2023). While those researchers focused primarily on comparing zero-shot LLMs against each other and the benchmark fine-tuned RoBERTa, this paper focuses on comparing the performance of different prompting methods to each other and fine-tuned RoBERTa.

As an alternate model to zero-shot and few-shot ChatGPT, I implemented DSPy, "a framework for algorithmically optimizing" LLM prompts (Khattab et al., 2023). DSPy's optimizers[3] are language-model driven algorithms that generate and tune prompts based on optimizing the model's performance on a metric and training dataset. Instead of the user tuning the prompt themselves, DSPy optimizers test numerous different prompts with the same base pipeline, and output the highest performing model based on the training set.

The datasets and models used in this paper are a subset of the datasets used in Shah and Chava (2023) due to time and cost constraints. Namely, due to lack of computational resources I will not be testing LLMs aside from ChatGPT, and I will be conducting less rigorous fine-tuning of RoBERTa. The choice to only test ChatGPT was due to an insufficient amount of RAM to load open-source models for testing, as well as the fact that Shah and Chava (2023) found ChatGPT to perform significantly better than other LLMs. Thus, ChatGPT will be able to sufficiently serve as our model for current high LLM performance, and allow us to see how changes in LLM prompting strategies compares to fine-tuned language models. All code and data used in this paper is available on GitHub.

Throughout this work I will try to answer the following research questions:

1. How does the performances of LLMs and fine-tuned models compare for financial domain tasks?

2. How does the performance of LLMs change with different prompting strategies (zero-shot, few-shot, algorithmically generating prompts) for financial domain tasks?

[1] https://github.com/mika-okamoto/zero-shot-finance/tree/main

[2] https://chat.openai.com/

[3] https://dspy-docs.vercel.app/docs/building-blocks/optimizers

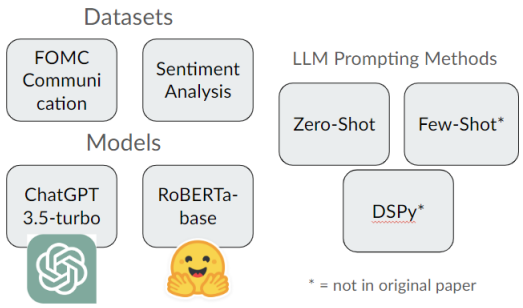

Figure 1: Overview of this paper's work.

2 Datasets and Tasks

I looked at Federal Open Market Committee (FOMC) communication hawkish-dovish-neutral classification from Shah et al. (2023), and financial sentiment analysis positive-negative-neutral classification from Malo et al. (2014). These were two of the datasets discussed in Shah and Chava (2023)

2.1 FOMC Communication

The monetary policy stance of the central banks is greatly influential on the market at large. The statements from FOMC have been found to have an extent on the market by various studies, including Shah et al. (2023). To capture this important natural language task in the financial domain, I evaluate the performance of our models on this dataset. In this dataset developed by Shah et al. (2023), sentences from the FOMC meetings, press conferences, and speeches are labeled as hawkish, dovish, or neutral.

2.2 Sentiment Analysis

Market sentiment has been seen to influence price movements, so sentiment analysis is an extremely popular NLP task in the financial domain. Hence, I evaluate our models' performance on it. This Financial Phrasebank dataset was developed by Malo et al. (2014) for financial sentiment analysis classification between positive, neutral, and negative sentiment. Like in Shah and Chava (2023), I only use the data where there is 100% annotation agreement.

3 Experiments

We run all the experiments for this paper ourselves and do not report any numbers from other works. I used the same 3 seeds to split datasets into train and test parts that the original paper used. In total, all ChatGPT calls cost $5 of OpenAI API credits.

3.1 Fine Tuning PLM

As our benchmark of how fine-tuned models can perform on these tasks, I used the base version of the RoBERTa (Liu et al., 2019) model. I used the same process as the original paper (Shah and Chava, 2023) to train this model, but pared down our grid search to only test 2 batch sizes (32, 16) and 2 learning rates (1e-4, 1e-5) due to limited computational resources. I chose to not test batch sizes 8 and 4 and learning rates 1e-6 and 1e-7 because they seemed to perform the worst on Shah and Chava (2023)'s results. I excluded training on RoBERTa-large due to running out of RAM when loading the model. I trained on Google Colab's free Google Compute Engine backend (GPU) and used PyTorch.

3.2 Zero-Shot with Generative LLMs

For zero-shot classification with ChatGPT, I used the original prompts listed in Shah and Chava (2023). I chose gpt-3.5-turbo [4] due to its speed, performance, and cheap price compared to gpt-4. I only used the test split for prompting, as zero-shot models do not require any training data.

3.3 Few-Shot with Generative LLMs

For few-shot prompting, I used the same model and base prompt used in the zero-shot experiment. Following the description of how to label the sentences, I inserted 9 labeled examples from the training set. These were evenly split between classes and randomly sampled from the same training set that the benchmark model was fine-tuned on. I also used ChatGPT 3.5-turbo and predicted labels on only the test set.

3.4 DSPy for Prompt Optimization

For implementing classification using the DSPy library, I used ChatGTP 3.5-turbo and a simple DSPy signature which instructed the model to classify between the three labels. I optimized using DSPy's MIPRO (Multi-prompt Instruction Proposal Optimizer), 30 examples from the training set evenly split between the classes, and 5 trials. With this, DSPy algorithmically found the best-performing prompt through using ChatGPT to generate prompts and then evaluating their performance on the training set. I then evaluated the optimized model's performance on the testing set.

[4] https://platform.openai.com/docs/models/gpt-3-5-turbo

Experiment	Mean Test F1	STD Test F1
Orig. RoBERTa	0.6990	0.0182
Orig. Zero-Shot	0.5837	0.0155
Repl. RoBERTa	0.6892	0.0030
Repl. Zero-Shot	0.5509	0.0114
Few-Shot	0.6049	0.0154
DSPy	0.6412	0.0286

Table 1: Results from tests with the FOMC Communication task and dataset from Shah et al. (2023).

Experiment	Mean Test F1	STD Test F1
Orig. RoBERTa	0.9735	0.0041
Orig. Zero-Shot	0.8929	0.0078
Repl. RoBERTa	0.9648	0.0097
Repl. Zero-Shot	0.8568	0.0127
Few-Shot	0.7792	0.0239
DSPy	0.8838	0.0143

Table 2: Results from tests with the Sentiment Analysis task and dataset from Malo et al. (2014).

4 Results

For benchmarking and evaluating the models and tasks previously discussed, I report the mean and standard deviation of the weighted F1 scores on the testing datasets. Unlike the Shah and Chava (2023) paper, there were no real cases where the model failed to follow the instructions and classify a piece of data, although there were instances where the output was given in a slightly different format than the expected. Those cases (for example, "Label: positive" rather than "positive") were transformed using text manipulation to extract the model's intended answer.

Results from both this replication (repl) and the original paper (Shah and Chava, 2023) (orig) in are given in Table 1 and Table 2.

4.1 Fine-Tuned PLM

My replication results for the fine-tuned RoBERTa base model were similar to the original paper's results. I believe the slight difference is due to my model being hyperparameter tuned on a subset of values that the original paper used and variance. As can be seen in both Table 1 and Table 2, RoBERTa performed the best out of all of the experiments. From this, we can gather that LLMs still have a way to go before they can surpass the performance of fine-tuned NLP models.

4.2 Zero-Shot

My replication results for Zero-Shot ChatGPT with chatgpt-3.5-turbo came out relatively similar to the original paper's results. I believe that the difference in results is due to ChatGPT's performance degradation over time (Chen et al., 2023), as the original paper ran these experiments 9 months earlier than I did. I did not change the prompt in any way, so the only explanation would be variance in the train/test split and ChatGPT's performance. This is of importance to note, as if ChatGPT continues degrading in performance, it will become less useful for these NLP tasks as time goes on.

4.3 Few-Shot

As can be seen in the tables, Few-Shot prompting with ChatGPT still performs worse than the original RoBERTa model. However, as can be seen in Table 1, it performs better than Zero-Shot ChatGPT on the FOMC Communication task, but as seen in Table 2, it performs worse than Zero-Shot ChatGPT on the Sentiment Analysis task. I believe that this means that Few-Shot prompting depends greatly on the qualities of the examples chosen, as I only gave 9 randomly sampled labeled examples from the training set. To optimize few-shot prompting as a strategy, there would likely need to be training to find the optimal examples to use in the prompt.

4.4 DSPy

Impressively, DSPy performed better than both Zero-Shot and Few-Shot ChatGPT by a decent amount for both datasets/tasks. I believe this suggests DSPy is worth looking into as a way of optimizing LLM prompts and pipelines for higher performance. With more optimization (more labeled examples given in the training set, more optimization rounds, etc), I believe DSPy can achieve even higher results than those I found. However, DSPy is costly, as it prompts the LLM many times during its optimization. Hence, I could not conduct further or more detailed tests as I ran out of OpenAI API credits.

These are some of the optimized prompts that DSPy found to perform well on its training set.

- FOMC Communication: Classify the sentence's stance on the monetary policy and its impact on the economy as hawkish, neutral, or dovish.

- Sentiment Analysis: Classify the sentence's

sentiment as either negative, neutral, or positive based on the financial performance of the mentioned Nordic company.

5 Conclusion

In conclusion, we compared ChatGPT's performance using different prompting strategies to our benchmark, the fine-tuned RoBERTa-base model. To answer our previously posed research questions, ChatGPT generally performs worse than RoBERTa for these financial domain tasks, but its performance varies greatly depending on prompting strategy. DSPy was found to perform the best, and will likely perform even better with more tuning. Further research could apply DSPy to other financial domain NLP tasks, or work on further tuning DSPy on these current tasks.

5.1 Challenges

The largest challenge during the course of this replication and expansion was getting around my limited resources. I had limited time (as I was in a solo group) and limited computational resources (as I do not have access to GPUs or the supercomputer cluster and thus had to rely on free resources). This made it difficult to run all of the tests that I wanted to run, especially running these experiments with more LLMs. I wish I could have had the resources to run these experiments with open source LLMs, and compared their performance to ChatGPT's performance. Other challenges I encountered included data not being in perfect format for dealing with, various programs crashing in the middle of experiments due to assorted reasons, and the difficulty of transforming these experiments built for zero-shot prompting to few-shot and DSPy. DSPy, in particular, was a completely different framework. Thus, the code all had to be completely rewritten, and not everything transferred perfectly.

5.2 Next Steps

Moving forward, I would suggest further testing ways to improve LLM performance on NLP tasks in the financial domain. I strongly believe that a strong combination of few-shot examples and tuned prompts will, one day, be able to surpass the performance of regular fine-tuned NLP models such as RoBERTa. I particularly think that DSPy shows great promise, and its capabilities should be researched and tested more going forward.

I would also like to test these experiments on other tasks / datasets, both inside and outside of the financial domain, and with other LLM models.

References

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is chatgpt's behavior changing over time?

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Pekka Malo, Ankur Sinha, Pyry Takala, Pekka Korhonen, and Jyrki Wallenius. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the American Society for Information Science and Technology*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver?

Agam Shah and Sudheer Chava. 2023. Zero is not hero yet: Benchmarking zero-shot performance of llms for financial tasks.

Agam Shah, Suvan Paturi, and Sudheer Chava. 2023. Trillion dollar words: A new financial dataset, task market analysis. pages 6664–6679.

A Appendix

A.1 Zero-Shot Prompts

I used the following prompts for the zero-shot experiments in this paper. These are taken directly from Shah and Chava (2023).

FOMC Communication: "Discard all the previous instructions. Behave like you are an expert sentence sentiment classifier. Classify the following sentence into 'NEGATIVE', 'POSITIVE', or 'NEUTRAL' class. Label 'NEGATIVE' if it is corresponding to negative sentiment, 'POSITIVE' if it is corresponding to positive sentiment, or 'NEUTRAL' if the sentiment is neutral. Provide the label in the first line and provide a short explanation in the second line. The sentence: {sentence}"

Sentiment Analysis: "Discard all the previous instructions. Behave like you are an expert sentence classifier. Classify the following sentence

from FOMC into 'HAWKISH', 'DOVISH', or 'NEUTRAL' class. Label 'HAWKISH' if it is corresponding to tightening of the monetary policy, 'DOVISH' if it is corresponding to easing of the monetary policy, or 'NEUTRAL' if the stance is neutral. Provide the label in the first line and provide a short explanation in the second line. The sentence: {sentence}"

A.2 Few-Shot Prompts

I used the same base prompts from the zero-shot experiments, with the following addition. Before the line "Provide the label ...", I insert "Examples: {examples}".

The examples are listed as follows: "Sentence: {sentence} Label: {label}. The sentences and labels (actual classes) are taken directly from the training set.

A.3 DSPy Initial Signatures

For the initial (pre-optimization) signatures for DSPy, I used the following prompts to specify the output classes and goal.

FOMC Communication: "Classify the sentence's stance on the monetary policy between hawkish, neutral, and dovish."

Sentiment Analysis: "Classify the sentence's sentiment between negative, neutral, and positive."