FinTrust Replication

Akaash Dash
Georgia Institute of Technology
adash37@gatech.edu

Sapan Patel
Georgia Institute of Technology
spatel794@gatech.edu

Aditya Natham
Georgia Institute of Technology
anatham3@gatech.edu

Max Zhao
Georgia Institute of Technology
qzhao305@gatech.edu

Abstract

Recent advancements in machine learning have started to impact many aspects of the world. The finance industry isn't an exception to this rule, as even a small edge in the ability to forecast prices can produce massive opportunities for profit, driving intense technological competition to find this edge. NLP in particular has been a great focus in financial forecasting and there has been a surge in models utilizing NLP to analyze documents, such as earnings reports, to predict a company's price. Despite these recent exciting advancements, the consistency of models, defined as the invariance under meaning-preserving inputs, is often not verified, which brings unreliability to these models. FinTrust is an evaluation tool that generates meaning-preserving copies of financial text, and was used in (Yang et al., 2023) to demonstrate the poor logical consistency of various NLP models. We seek to replicate the experiments conducted with FinTrust and later extend the tool to allow automated consistency evaluations of various models. Resources are available at https://github.com/sapanpatel3775/FinTrust_Replication.

1 Introduction

Recently, Natural Language Processing (NLP) models have been increasingly used for forecasting tasks in the financial domain. Despite this growth, there has been considerable skepticism regarding the trustworthiness and robustness of these models, important factors in a domain with high requirements such as finance. To combat this, research has emerged to evaluate model robustness. From this, one such means has shown to be promising: causal explanation. Causal explanation involves exploring the cause-and-effect reasoning embedded within models. However, causal explanation has some underlying requirements. One of such is consistency – a model's ability to generate consistent outputs with different inputs carrying the same semantic meaning. While consistency may not explicitly demonstrate causal explanation, it underscores the presence of causal reasoning. In the financial domain, there has been a focus on financial results but a notable lack of analysis concerning the underlying reasoning. Some existing work has explored evaluating implicit preferences, while the sole study on consistency, namely "FinTrust," primarily aims to explore a holistic measure for stock movement prediction, incorporating consistency as a criterion of trustworthiness (Yang et al., 2023). However, despite the stated objective, the paper falls short of fully achieving this goal. Nevertheless, it does conduct evaluations on consistency. Hence, the purpose of this paper is to replicate the consistency measuring features observed in the "FinTrust" paper.

2 Original Paper

The original paper aims to develop a comprehensive measure for stock movement prediction by integrating consistency as a criterion of trustworthiness. However, it does not effectively stick to this objective. The introduction of FinTrust, a tool for measuring consistency in Pretrained Language Models (PLMs), lacks substantial explanation and definition within the paper, along with insufficient code documentation. The paper outlines three main goals: assessing implicit preferences in PLMs, measuring the accuracy of stock movement prediction using real-world earnings call data after meaning-preserving modifications, and proposing a trading simulation. However, the focus shifts away from the first goal, which we believe to be the most important. Our research aims to replicate and expand on this aspect by evaluating implicit preferences in PLMs, consistency in PLMs, consistency in fine-tuned PLMs, and consistency in financial forecasting models. To begin, we define consistency as the stability of model outputs or predictions under meaning-preserving transformations, including negation, symmetric, additive, and transitive transformations.

Negation: the ability of a model to generate converse predictions for texts with opposite meanings

- Example: replace a word with its antonym

Symmetric: property of a model where the order of the inputs does not affect the output

- Example: reverse the order of a sentence

Additive: property of a model to predict the stock movement based on the combination of two inputs that share the same label

- Example: Combine pairs of inputs that share the same label

Transitive: the ability of a model where the perceived sentiment of a company should be reflected in the performance of the top-valued company in the same industry

- Example: Replace the company name within the same sector

To evaluate models, we refer to the paper's methods, which define two setups. The first involves mask token filling to generate model predictions on positive and negative sentiments to evaluate implicit preferences, a well-defined process. The second setup moves on to stock prediction tasks and trading simulation tasks. While the stock prediction tasks offer valuable insights for measuring consistency in forecasting models, the trading simulation task is relatively trivial.

A significant issue lies in the clarity of the results presented for the forecasting model evaluation. The paper initially presents a predictive analysis of the results before and after fine-tuning on a consistency-transformed dataset. While this serves as a good demonstration of the dataset's application, it fails to provide meaningful insights into the actual consistency of the models. Later in the paper, a consistency evaluation for the forecasting models is provided, but it is not clearly stated whether these results are from before or after the fine-tuning process. Additionally, there is a lack of a direct comparison between the consistency measures before and after fine-tuning for these models. This comparison would have offered valuable insights into the consistency of financial forecasting models and how it is impacted by the fine-tuning process.

3 Replication Idea

The primary objective of this paper is to conduct an evaluation of Pretrained Language Models, focusing on implicit preferences and consistency. Implicit preferences within PLMs are a broad range of possibilities, including biases towards specific stocks or factors, as well as semantic biases. These biases may arise due to training data bias or the incorporation of human semantic bias during the model's training phase. For instance, implicit preferences might manifest as a tendency for the model to favor stocks from certain industries or sectors over others, reflecting underlying biases present in the data used for training. Additionally, semantic biases could emerge from societal or cultural norms embedded in the language data, leading to preferential treatment of certain entities within the model's output.

Consistency, another focus point of this study, holds importance in financial domains. Public speakers representing companies often use various rhetorical strategies, such as euphemisms and theatrics, but the underlying meaning of their messages remains the same. By using logical consistencies, we aim to simulate such scenarios and see whether a model can determine the actual meaning despite rhetorical changes. This not only displays the robustness of PLMs, but also shows their capacity for causal reasoning, giving us understanding of their abilities in practical applications.

4 Data

The original paper provides several datasets for analysis. It provides a dataset of earnings call transcripts, and a dataset with the transcripts merged with corresponding stock price movements, used to evaluate forecasting models. The paper then provides a subset of this merged data with logical transformations applied. The paper also references earnings call transcripts from a previous study (Qin and Yang, 2019). The usage isn't very clear, but we suspect it is used to fine-tune PLMs before further evaluation.

5 Models

For PLMs, the paper lists several for evaluation: BERT-Base, BERT-Large, RoBERTa-Base, RoBERTa-Large, DistilBERT, and FinBERT. However, the paper references two FinBERT models. The first model is strictly a sentiment analysis model, and gave nonsensical outputs in limited mask filling testing (Araci, 2019). Because of this, we use the other FinBERT model, which is more suited for the tasks (Yang et al., 2020b). All the mentioned models are readily available on HuggingFace.

6 Replication Methodology

6.1 Evaluating Implicit Preferences in PLMs

To evaluate implicit preferences in PLMs, we mask semantic tokens and evaluate model's prections of the masked tokens.

We first compile a comprehensive list of positive and negative semantic words, drawing from the negation consistency script available in the original repository. We do this because it is what the original authors used to determine negative and positive sentiments, and also will form the basis of the negation consistency.

We then segment each earnings call transcript into individual sentences and proceed to mask specific words according to the compiled lists. For sentences containing multiple words from either list, we iteratively apply the masking process to each word. Sentences without words from either list are excluded. After the masking phase, each PLM is used to fill the masked tokens, and the predictions are recorded.

These predictions are categorized into labels: exact prediction matches, same sentiment predictions, opposite sentiment predictions, or none, which can be words with no sentiment or undetermined tokens. With the labels, we conduct a comprehensive analysis.

6.2 Evaluating Consistency in PLMs

To evaluate consistency in PLMs, we give the PLMs a sentence from the earnings call and ask it to predict if the stock will go up and down based on the sentence. We then conduct meaning-preserving logical transformations on the data and see if the predictions remain the same.

First, we create a prompt template that will be used to enclose the sentence and ensure the model gives an up/down prediction. We then query the model individually for each sentence of all earnings calls based on the template and record the results. Next, logical transformations are applied to the sentences using the scripts provided by the original authors. After that, we query the models again and compare if the predictions remain the same or change.

The prompt template used was:

```
Given the following text from an earnings
call:

[PHRASE]

Based solely on the information provided
in this text, do you predict the
stock price for the associated company
will go up or down in the near future?

The stock price for the associated company
will likely go [MASK].
```

6.3 Evaluating Consistency in Fine-Tuned PLMs

To evaluate consistency in fine-tuned PLMs, we use a similar methodology to the one previously described, but first fine-tune the PLMs.

The earnings call dataset reference in the paper provides three earning call transcripts (Qin and Yang, 2019). We use these to fine-tune PLMs using a standardized pipeline. After this, we use the same approach done before.

6.4 Evaluating Consistency in Financial Forecasting Models

To assess the consistency of forecasting models, we compare predictions on various time window predictions before and after logical transformations are applied.

First, we partition our merged earnings call and stock value dataset into distinct train, validation, and test sets, maintaining the 7:1:2 (train:validation:test) ratio stated in the original paper. We then use the logical transformations to create four additional copies of each sample in each set, one for each transformation. Next, we train the forecasting models using this data. Finally, we generate predictions on the directional movement of stocks—either upward or downward—over various time intervals (3, 7, 15, and 30 days).

We then apply logical transformations to the test set and generate predictions in a similar manner. The results come from a comparative analysis assessing whether the predictions generated from the transformed test sets align with or diverge from the original predictions.

7 Results and Discussion

7.1 Evaluating Implicit Preferences in PLMs

The label results from the mask filling provides a vast amount of information on the consistency of models.

We find that for all models, a large portion of predictions fall into the none label. This shows that these models are overall not very good for financial NLP tasks in general. However, the RoBERTa models and FinBERT had significantly less none labels when compared to the other models. This shows that these models are better at understanding context and NLP tasks overall. The lower proportion of none labels in FinBERT is specifically interesting because it is based on the BERT model, but performs better than other BERT or BERT based models. This could be due to the familiarity with financial contexts and indicates capability in financial tasks. The large proportion of none results significantly skew any further analysis, so we disregard them in any further calculations. The following results can be seen in Table 1.

PLM	Params	Neg	Pos	Consistency
BERT-base	110M	+	+	91.93%
BERT-base	110M	+	-	77.50%
BERT-base	110M	-	+	95.50%
BERT-large	340M	+	+	92.71%
BERT-large	340M	+	-	80.37%
BERT-large	340M	-	+	95.66%
RoBERTa-base	125M	+	+	95.34%
RoBERTa-base	125M	+	-	85.15%
RoBERTa-base	125M	-	+	97.52%
RoBERTa-large	355M	+	+	**96.38%**
RoBERTa-large	355M	+	-	**89.14%**
RoBERTa-large	355M	-	+	**97.98%**
FinBERT	110M	+	+	94.79%
FinBERT	110M	+	-	81.92%
FinBERT	110M	-	+	97.51%
DistilBERT	66M	+	+	91.05%
DistilBERT	66M	+	-	80.83%
DistilBERT	66M	-	+	93.45%

Table 1: The results of the implicit preference measurement in PLMs via masked token predictions, splitting by negative and positive token predictions. '+' denotes that the attitude of the word with the specific polarity will be predicted while '-' means that we do not consider tokens with a specific polarity.

We notice that the models had relatively high consistency, greater than 90%. This is in contrast with the original paper, which claimed that all models had relatively low consistency that would impact performance in financial requirements. We also see that the consistency is much lower when predicting negative sentiments as opposed to positive sentiments, indicating some bias towards negative sentiments. This is in line with the original paper, which found the same outcomes.

When looking at models comparatively, we find several patterns in line with the original paper. The consistency score of FinBERT is significantly higher than the other BERT models, showing proficiency in financial contexts. Additionally, we can observe that increasing the size of models leads to higher consistency. This is shown in the performance of BERT-base vs BERT-large and RoBERTa-base and RoBERTa-large. This pattern may be specific to the BERT family, but additional experiments would be needed to validate that claim. Of all the models, RoBERTa-large had the highest consistency scores in all categories and DistilBERT had the lowest overall consistency. All of these findings are shared with the original paper.

Overall, we had higher consistency scores than the paper, but the patterns found were the same, indicating significant results in the results of this portion of the paper.

7.2 Evaluating Consistency in PLMs

When evaluating PLMs for transformational consistency, we encounter limited success and even some task difficulty. The results can be found in Table 2. Note that negation predictions are flipped when compared as they are expected to be the opposite of the original predictions. Also note that predictions are not compared to actual stock movement, just the original prediction before transformations to ensure consistency.

PLM	Neg	Sym	Tra	Add	AVG
BERT-base	11.47	87.08	90.29	61.31	62.54
BERT-large	0.17	99.83	99.91	99.45	74.84
RoBERTa-base	2.91	97.77	98.24	93.41	73.08
RoBERTa-large	22.40	92.39	89.88	70.72	68.85
FinBERT	8.31	94.27	92.15	57.16	62.97
DistilBERT	0.00	100.0	100.0	100.0	75.00

Table 2: The results of the consistency measurement in PLMs via masked token predictions. (In %)

We find several observations when looking directly at the models. First, DistilBERT has 100% accuracy except negation consistency, with 0%, indicating that it gave the same output each time. This shows the model having difficulty with the task, which could be a consequence of the prompt template chosen. However, it could also show the weakness of DistilBERT, which was previously shown in the first task.

We also notice that BERT-large had the same tendency to output the same token the majority of the time, however this model had some deviation and was not as egregious as DistilBERT. RoBERTa-base shows a similar output pattern, but it is not as clear and obvious, meaning that the prediction results could be valid.

When looking at individual transformations, we find that all models struggled with the negation consistency. The best performing model, RoBERTa-large, only had a 22.40% accuracy. Because negation consistency consists of similar amounts of both positive and negative negations, it is unlikely that this is due to a bias in the prompt. More likely, models read into other factors of the text such as diction or phrasing as opposed to a key word sentiment.

Models also did not perform as well with the additive task, which is surprising. It is expected that with two phrases of the same sentiment, the model would easily be able to conclude with the overall sentiment, but this is not reflected.

Despite having a low average consistency, RoBERTa-large seems to be the most balanced across the transformations. It is difficult to pick a single model as 'best' because of tendency of some models to produce the same output despite transformations.

Also a noticed trend, but not one reflected in the chart, is FinBERT struggling to produce sensible outputs. All the other models correctly had outputs of either 'up' or 'down', but FinBERT had many instances where the predicted token was a punctuation mark or obscure word.

7.3 Evaluating Consistency in Fine-Tuned PLMs

The results for consistency evaluation after fine-tuning can be seen in Table 3.

PLM	Neg	Sym	Tra	Add	AVG
BERT-base	46.61	53.50	53.51	52.64	51.57
BERT-large	22.35	79.69	79.29	78.13	64.86
RoBERTa-base	48.43	52.23	52.52	50.92	51.03
RoBERTa-large	44.41	58.84	59.82	57.49	55.14
FinBERT	46.33	54.30	54.58	51.49	51.68
DistilBERT	0.60	99.44	99.36	99.39	74.70

Table 3: The results of the consistency measurement in fine-tuned PLMs via masked token predictions. (In %)

After fine-tuning, almost all models seem to perform significantly differently. The one exception to this is DistilBERT, which still produces identical tokens the majority of the time. We also find that BERT-large and RoBERTa-base no longer show a similar pattern to DistilBERT.

The most obvious shift is the improvement in the negation consistency. All models significantly improved, with the best performing model performing at 48.43%. Interestingly, the model that performed the best at negation consistency is the not same model that performed the best before fine-tuning, but is RoBERTa-base.

However, all models (except DistilBERT) performed significantly worse at all other consistency measurements. In fact, it appears that all consistency measurements appear to tend towards 50%, which is no better than a coin flip. This indicates that the models are struggling to predict movement based on the given context and are producing guesses for each query. This is almost a degradation of performance from before fine-tuning, which is very surprising.

7.4 Evaluating Consistency in Financial Forecasting Models

Results for consistency measurement in the listed financial forecasting models were unobtainable due to the numerous difficulties encountered with these models and the unclear processes used to run them. The paper did not provide a clear explanation of how to run the models, and the necessary code was not provided, further complicating the replication process. More details on these challenges are discussed in the challenges section.

8 Challenges in Replication

The original paper claimed to have code on GitHub, but the paper's repository only contained code for generating logical transformations. Because of this, we have to generate the rest of the code from scratch following the methods listed in the paper. This can be challenging at times, as many concepts in the original paper are ambiguous or poorly detailed.

With respect to the forecasting models, they have proven to be very complicated to figure out how to load and run. The paper contains no information on what settings or parameters are used, what format the data is presented in, or any details on how to run any of the models. Additionally, three of the models were subjects of replication studies by other groups: Event (Ding et al., 2015), HTML (Yang et al., 2020a), and MRDM (Qin and Yang, 2019). We reached out to these groups but did not have much success in getting responses or finding information on how to make them work. The last used model was XGBoost (Chen and Guestrin, 2016), but this model is purely numerical, meaning that the text would have needed to be converted into encodings to be able to be passed to this model. There are no details at all about encoding text or what data was used for XGBoost, making it impossible to replicate this model.

9 Future Directions

Future research could explore experimenting with different cloze-style prompt templates. In this study, we tried several different templates due to the initial ones exhibiting bias towards either upward or downward predictions. The current prompt may still contain biases, and there may be ways to further mitigate this issue. A comparison of different prompts would allow us to understand how to prompt models for forecasting without biasing predictions in any particular direction (Xu et al., 2024).

Another avenue for future work is using consistency-transformed datasets for training PLMs before evaluation. This approach will enable a comparison between no fine-tuning, plain earnings call fine-tuning, and transformed earnings call fine-tuning, providing insights into the impact of consistency-aware training on model performance. Additionally, creating a clarifying consistency evaluation for a wide range of forecasting models and expanding on this part of the methodology could yield valuable results.

Recent papers suggest that fine-tuning may be less effective than prompting with in-context learning, and the poor fine-tuning results of both this paper and the original paper support this idea (Lin et al., 2023). Research into using in-context learning for consistency could lead to promising outcomes. Experiments with few-shot and zero-shot learning may also offer more opportunities for exploring consistency in these learning paradigms (Shah and Chava, 2023; Phogat et al., 2023).

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16. ACM.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *International Joint Conference on Artificial Intelligence*.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning.

Karmvir Singh Phogat, Chetan Harsha, Sridhar Dasaratha, Shashishekar Ramakrishna, and Sai Akhil Puranam. 2023. Zero-shot question answering over financial documents using large language models.

Yu Qin and Yi Yang. 2019. What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 390–401, Florence, Italy. Association for Computational Linguistics.

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

Ziyang Xu, Keqin Peng, Liang Ding, Dacheng Tao, and Xiliang Lu. 2024. Take care of your prompt bias! investigating and mitigating prompt bias in factual knowledge extraction.

Linyi Yang, Yingpeng Ma, and Yue Zhang. 2023. Measuring consistency in text-based financial forecasting models.

Linyi Yang, Tin Lok James Ng, Barry Smyth, and Ruihai Dong. 2020a. Html: Hierarchical transformer-based multi-task learning for volatility prediction. In *Proceedings of The Web Conference 2020*, pages 441–451.

Yi Yang, Mark Christopher Siy UY, and Allen Huang. 2020b. Finbert: A pretrained language model for financial communications.