Conversational AI for Market Intelligence: Learning the Grammar of Financial News to Optimize Portfolios

Mo Afkhami, *Ph.D. in Finance, CFA*
Financial Services, Blend

January 2025

Summary

This paper proposes a novel approach to financial forecasting that leverages Large Language Models (LLMs) not just for time-series pattern detection, but for interpreting the impact of new unstructured information on asset prices—a crucial distinction from previous methodologies. With the introduction and widespread usage of Large Language Models (LLMs) in recent years, several studies have focused on leveraging the power of LLMs for time series analysis, particularly in asset price prediction. However, similar to previous attempts at implementing complex deep learning architectures over the past decade, initial reports suggest only modest success. This is primarily because asset prices are predominantly reactive to external factors and information, which means that detected past patterns are not strong predictors of future trends. One conclusion from this observation is that, rather than solely leveraging LLMs to detect patterns, it would be more effective to utilize LLMs' capability in handling unstructured information to interpret the effects of information on prices. This idea forms the core of the proposal presented in this document. We propose two methods to leverage LLMs in conjunction with a deep learning architecture to estimate the impact of new information on asset prices.

The output of this model can serve as input for several investment-making decisions. The estimates can serve as real-time CMEs for products focused on portfolio optimization or goal-based investing, or as discussed in the document, directly be used to create an optimal portfolio. In addition, the model can serve as a POC for a more ambitious project, where a portfolio's weights would be automatically updated with the arrival of new unstructured data.[1]

Introduction

The Efficient Market Hypothesis (EMH), in its semi-strong form, suggests that all past information and presently public available information are reflected in current prices [3]. When the hypothesis is true, for a person with no access to material non-public information, prices are Markov processes. i.e., all there is to know about the price in the next time step is that it is the current price plus some white noise. But how efficient are the securities markets in reality? In his seminal work, reviewing several strategies employed by hedge

[1] The details of that project are outside the scope of this proposal.

funds, and smart money, Pederson [4] contends that markets are neither perfectly efficient nor completely inefficient. Rather, they are inefficient enough that money managers can be compensated for their costs through the profits of their trading strategies. Furthermore, while several studies [5, 1] have confirmed that the weak form of EMH (i.e., prices reflect all past information) holds in large securities' markets, the semi-strong form has proven to be more controversial. In other words, the reaction of the market to new piece of information is not always instantaneous nor with the 'right' magnitude. These observations encourage us to explore the *predictability* of prices.

This document proposes two novel methodologies, in high and low latency environments, for leveraging Large Language Models (LLMs) for modeling time series and in particular, to predict the first and second moment of securities, fixed income products and macroeconomic variables. To date, the use of LLMs in the prediction of financial assets has shown only modest improvements compared to traditional econometric methods or deep learning architectures [2]. In addition, the relatively mediocre performance of these models in the securities' markets suggest that more complex methods that rely on detecting past trends to identify the future are unlikely to yield better results. However, we believe that by efficiently harnessing the capabilities and strengths of LLMs, the proposed method has the potential to significantly outperform previously reported results.

In the first methodology, we focus on highly liquid stocks within a low frequency trading environment (e.g., updating investment in a particular security based on its quarterly earnings report.), and aim to model the probability distribution of a security's returns following an earning report. We then expand this approach to an stochastic optimization environment, where an agent updates an investment portfolio of n stocks and cash at each earnings announcement, with the goal of maximizing profits over time.

In the second methodology, we switch our focus to higher-frequency trading. Using a Sequence-to-Sequence Model (Seq2Seq) approach with an encoder-decoder Transformation model predict the impact of textual information as an incoming sequence on the sequence of prices of securities that make our universe of investments.

Once the core idea is established, we extend this framework by incorporating market asymmetry, recognizing that financial markets tend to react more sharply to negative news than to positive news. By explicitly modeling these asymmetric reactions, we develop a reinforcement learning-based portfolio rebalancing policy that dynamically adjusts asset allocations in response to sentiment-driven market shifts, ensuring a more risk-aware and adaptive trading strategy in volatile environments.

Conceptual Framework

Conversation Paradigm: The interaction between news and time series data is viewed as a dialog in which news articles provide context or stimuli and time series data (such as stock prices, commodity prices) react accordingly. This approach facilitates a dynamic understanding of how external information influences market trends over time. It treats financial markets as an adaptive system where new information acts as an exogenous stimulus, and market movements serve as structured responses. The model learns the underlying "grammar" of this interaction, allowing it to anticipate not just direct price changes, but the broader reaction function of asset classes to emerging financial narratives

Grammar Learning: An LLM is employed to learn the "grammar" or patterns of these interactions. The model is trained to recognize how different types of news events impact time series trends, capturing the intricate dependencies and nuances within these relationships.

Visualizing the Conversation – Attention Heatmap : To further illustrate how the transformer model learns the grammar of financial narratives, Figure 1 presents an attention heatmap that highlights which words in a financial news event receive the most focus when predicting market movements. Each row and column represent a word or phrase, and the color intensity reflects the degree of attention assigned by the model.

For example, in a news headline such as"The Fed raised interest rates by 0.5%, causing market volatility", the model assigns higher attention weights to key financial terms like "Fed", "interest rates", and "market volatility", while discounting less relevant words like "The" or "by". This demonstrates that the model is not just memorizing past patterns but is actively learning the structural relationships between financial events and asset price movements. This visualization provides interpretable insights into how financial markets

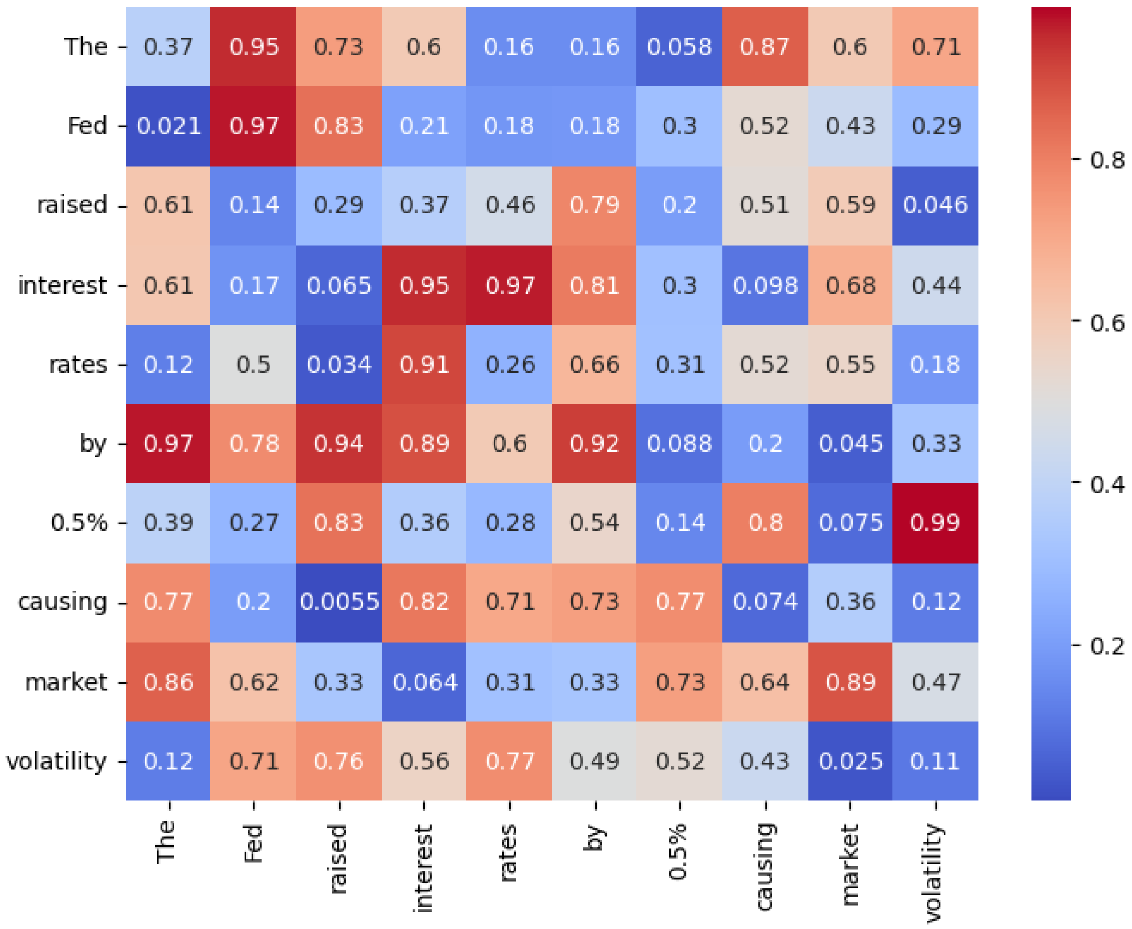

Figure 1: A sample heatmap of attention interpretation

process new information, reinforcing the validity of model-driven trading strategies. By analyzing these attention distributions across multiple news events, we gain a more granular understanding of market sensitivity to external narratives, enabling more robust risk-aware trading strategies.

Data Required and Feature Engineering

Some of the main data sources are:

1. News, earning, and financial reports
2. Historical data of security prices
3. FRED and other macro-economic data.

Other data sources such as social media sentiment and US senators' portfolio composition[2] can also be added to the model.

In addition, to the standard feature engineering, the feature engineering would require several additional steps, such as:

- **Text Embeddings:** Utilize an LLM to generate embeddings for the news articles. These embeddings capture the semantic meaning and contextual nuances of the news content.
- **Sequential Representation:** Represent both the news data and the time series data as sequences to capture their temporal dependencies and interactions. This aids in modeling how news events influence time series trends over time.
- **Aggregation:** The logic behind aggregation would be determined based on the accuracy This is particularly applicable to variables that represent "sentiment". Rather than employing for example all social media sentiment on a stock between two time steps, we create aggregate measures or scores to capture this information.

Low Frequency Information

The main assumption behind this approach is that while EMH may hold in its weak form, it is not valid in its semi-strong form. In other words, it takes some time for new information to be reflected in prices. Our primary goal with this approach is to observe and measure the impact of earnings reports on a stock's return. i.e., $P_R(r_{t+1} \mid E_t)$. Where P_R is the PDF function of returns, r_{t+1} is the return in the next time step, and E shows the earning report content.

Implementing this approach would require converting the content of the earnings reports to vector embeddings. The historical embeddings, along with additional features (such as macro variables, and historical prices) would then be used to estimate the reaction of the stock price (returns) to the news. In mathematical terms, the return of stock i in the next time step, $r_{i,t+1}$, can be explained as:

$$r_{i,t+1} = \hat{f}(E_{i,t}, E_{j,t}, r_{i,t}, \alpha_t) \quad \forall i, j \in \{1..n\}$$

In the universe of n stocks, with α_t capturing the macro factors. A deep learning Architecture can be used to train the model. Though tree-based methods such as XGBoost

[2]This data is publicly available. See for example https://senatestockwatcher.com/api

and LightBGM may also be worth expoloring. The model can be trained on all the stocks with timely quarterly reports and 10-Ks, using a reasonable time frame. This model would serve as the first building block in our work.

Creating The Optimal Portfolio

With the estimated probability distribution of returns, we can now construct an optimal portfolio that updates dynamically. The optimization process follows a stochastic rebalancing approach, adjusting allocations based on new market information.

Specifically, an agent managing n stocks and cash continuously re-evaluates portfolio weights at each decision point (e.g., an earnings report release). This allows for adaptive reallocation based on newly updated risk-return expectations

We propose two solution approaches to this problem. In the first, we frame the problem as a sequential decision-making task, where an agent dynamically adjusts portfolio weights based on new unstructured financial information. Borrowing from the logic of Thompson sampling in the multi-arm bandit literature, we optimize allocation by continuously updating our belief distributions about asset returns—a reinforcement learning-inspired framework. The second solution, resembles a modern portfolio optimization problem.

In the first approach, once we have the estimated probability distribution of each stock, at each point in time, for each dollar of of the total wealth W_t, we take one sample from every distribution,

$$\hat{r}_{i,t} \sim P_i(r)$$

and we assign that one dollar to the stock with the highest return:

$$i^* = \arg\max_i \hat{r}_{i,t}$$

Resulting in the following allocation:

$$A_{i,t}^{(d)} = \begin{cases} 1 & \text{if } i = \arg\max_i \hat{r}_{i,t} \\ 0 & \text{otherwise} \end{cases}$$

thus, the percent allocation in each stock can be expressed as:

$$a_{i,t} = \frac{1}{W_t} \sum_{d=1}^{W_t} A_{i,t}^{(d)}$$

Figure 2 illustrates the sampling process.

When to update the policy: We assume earning reports arrive on a quarterly basis but at different times for different stocks. Each of these points would serve as a decision time step. However, additional time steps can be included, based on every point where a new data arrives.

In the second approach, the expected return of the portfolio of n assets at each point in time is

$$\mathbb{E}[R_t] = \mathbb{E}\left[\sum_{i=1}^{n} a_i r_i\right] = \sum_{i=1}^{n} a_i \mathbb{E}[r_i]$$

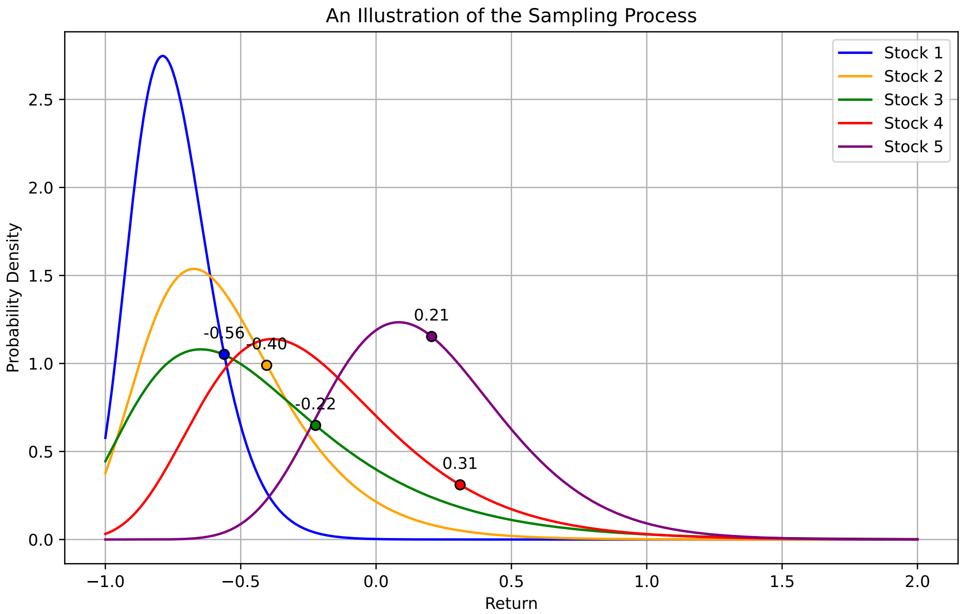

Figure 2: Sampling from five return distributions

with the variance of the portfolio being:

$$\mathrm{Var}(R_t) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \mathrm{Cov}(r_i, r_j)$$

Thus, the goal is to maximize:

$$\max \quad \sum_{i=1}^{n} a_i \mathbb{E}[r_i] - \frac{\lambda}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \mathrm{Cov}(r_i, r_j)$$

where λ captures the risk aversion factor of the investor and the problem is subject to the usual constraints such as the sum of the weights being equal to one.

High Frequency Information

In a higher frequency of trade and with low latency data coming in, the prediction and consequently the asset allocation problem become slightly different. Here, instead of modeling what would be the impact of an earning report on a specific stock, we are interested in modeling the impact of the flow of the information on a sequence of stocks at each point in time. i.e.,

$$(r_{1,t+1}, r_{2,t+1}, ..., r_{n,t+1}) = \hat{f}(\mathrm{news}_t)$$

The relation or the the grammar between pieces of information and prices can be modeled using a Seq2Seq model that aims to learn the interaction grammar between information and return data. In the workflow, a new piece of information comes in, and gets embedded as a vector. This vector is then used as an input sequence to estimate the output sequence of stock returns.

Unlike traditional autoregressive models, the encoder-decoder architecture of the Seq2Seq Transformer allows the model to flexibly encode financial news events of varying lengths (from short headlines to full earnings reports) while predicting their impact on time-dependent asset price trajectories. This enables more robust event-driven forecasting where the response time of markets varies across different asset classes. The encoder processes the input sequence (news embeddings) and transforms it into a fixed-size context vector, which the decoder then uses to generate the output sequence (stock returns). We propose using transformers for both encoders and decoders.

A key advantage of leveraging Transformer-based models for financial forecasting is their ability to capture asymmetric financial reactions to news events. Traditional econometric models often assume a symmetric response to positive and negative information, but financial markets are known to react more sharply to negative shocks. Our approach dynamically weighs the impact of news events using a self-attention mechanism, ensuring that large downside risks are properly accounted for while capturing slower, more gradual positive price movements. This makes the model particularly well-suited for volatility forecasting, crisis detection, and sentiment-driven asset allocation

Once the predictor model is developed, an asset allocation algorithm similar to the one described in the previous section can be used to create optimal portfolios.

Asymmetric Financial Reactions and Portfolio Optimization

Markets exhibit asymmetric reactions where negative news often has a larger and more immediate impact on asset prices than positive news. Traditional models fail to capture this, assuming symmetric return distributions. To address this, we introduce an asymmetry-aware reinforcement learning framework, where an agent dynamically adjusts portfolio weights based on the sentiment and impact of financial events. Using a modified Sharpe ratio objective with a downside risk penalty, our approach ensures greater sensitivity to market downturns, optimizing allocations under shifting conditions. Furthermore, we extend Thompson Sampling to incorporate asymmetric risk weighting, allowing for dynamic portfolio rebalancing that prioritizes resilience in volatile markets.

Asymmetric Reactions

Negative news typically induces a stronger and more immediate reaction than positive news. This asymmetry can be mathematically represented as:

$$P(r_t|\text{positive news}) \neq P(r_t|\text{negative news})$$

where:

- $P(r_t)$ represents the conditional probability distribution of returns at time t.
- positive news refers to news events that generally have an upward expectation on asset prices (e.g., strong earnings reports, favorable macroeconomic indicators).
- negative news includes events likely to trigger downward market movements (e.g., earnings misses, regulatory crackdowns, macroeconomic distress).

This asymmetric response necessitates a modified portfolio optimization strategy that accounts for risk asymmetry in market reactions.

Asymmetry-Aware Reward Function

Traditional reinforcement learning-based portfolio optimization often optimizes the Sharpe ratio by maximizing expected return relative to risk. The standard reward function is given by:

$$R_t = \frac{\mathbb{E}[r_t] - R_f}{\sigma_t}$$

However, this formulation assumes symmetric risk exposure, meaning that both positive and negative deviations from expected return are weighted equally. To incorporate asymmetric downside risk, we introduce a penalty function:

$$R_t^{\text{asym}} = \frac{\mathbb{E}[r_t] - R_f}{\sigma_t} - \lambda \max(0, -r_t)^p$$

where:

- R_t^{asym} is the adjusted portfolio reward incorporating asymmetric risk aversion.
- λ is a downside risk penalty coefficient that controls how much the model penalizes negative returns.
- p is a power parameter (e.g., quadratic penalty for $p = 2$) that intensifies penalties on extreme downside risks.
- $\max(0, -r_t)$ ensures that the penalty applies only to negative returns.

This adjusted reward function biases the model towards more conservative allocations in high-risk environments, ensuring that downside volatility has a greater impact on decision-making than upside deviations.

Asymmetric Thompson Sampling for Portfolio Rebalancing

To optimize portfolio weights dynamically, we employ Thompson Sampling, a Bayesian approach commonly used in multi-armed bandit problems. Standard Thompson Sampling selects an action by drawing from estimated return distributions:

$$r_i \sim P_i(r)$$

where:

- r_i is the sampled return for asset i.
- $P_i(r)$ is the estimated probability distribution of returns for asset i.

To incorporate asymmetry, we modify the sampling distribution to downweight high-risk assets under negative market sentiment:

$$P_i^{\text{adj}}(r) = P_i(r) - \lambda \max(0, -r)^p$$

where:

- $P_i^{\text{adj}}(r)$ is the adjusted probability distribution that incorporates downside risk adjustments.
- The penalty term $\lambda \max(0, -r)^p$ ensures that assets with high downside risk are assigned lower probabilities in the sampling process.

Using this adjusted distribution, portfolio weights are assigned using a softmax function:

$$w_i^t = \frac{e^{\beta P_i^{\text{adj}}(r)}}{\sum_{j=1}^{N} e^{\beta P_j^{\text{adj}}(r)}}$$

where:

- w_i^t represents the weight assigned to asset i at time t.
- β is a risk-sensitivity parameter that controls how much risk-adjusted returns influence allocations.
- The softmax function ensures that weights sum to 1, making it a valid probability distribution over asset allocations.

Policy Gradient Optimization

Portfolio optimization under asymmetric risk can also be formulated as a reinforcement learning problem, where an agent learns an optimal allocation policy using a reward-maximizing strategy. The objective function is:

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t|s_t) \cdot R_t^{\text{asym}}\right]$$

where:

- $J(\theta)$ is the expected cumulative reward function parameterized by policy weights θ.
- $\pi_\theta(a_t|s_t)$ is the policy function that determines portfolio allocation actions a_t given the current market state s_t.
- R_t^{asym} is the asymmetry-adjusted reward function, ensuring that the policy prioritizes downside risk management.

The reinforcement learning agent updates its policy iteratively to optimize allocations, using:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

where:

- α is the learning rate, controlling the speed of policy updates.
- $\nabla_\theta J(\theta)$ is the gradient of the asymmetric risk-aware policy function.

Final Portfolio Construction Algorithm

Based on the above formulations, our asymmetric portfolio optimization strategy follows these steps:

1. Extract financial news sentiment embeddings using a pre-trained Large Language Model (LLM).
2. Estimate return distributions $P_i(r)$ for each asset based on historical price movements and sentiment-adjusted factors.
3. Adjust distributions using asymmetry-aware Thompson Sampling:
$$P_i^{\text{adj}}(r) = P_i(r) - \lambda \max(0, -r)^p$$
4. Compute updated portfolio weights using:
$$w_i^t = \frac{e^{\beta P_i^{\text{adj}}(r)}}{\sum_{j=1}^{N} e^{\beta P_j^{\text{adj}}(r)}}$$
5. Optimize portfolio allocations using reinforcement learning, updating policy weights as:
$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

Tuning and Model Performance Measurement

A conventional regressor loss function such as MSE can be used to train the low frequency and the high frequency (Seq2Seq) model. The train-test split should be done using cut-off dates. The model's performance should be benchmarked against the existing econometric and deep learning models as well as the LLM based approaches.

Simulation

Beyond prediction, the model can also be leveraged for scenario simulation, enabling the generation of synthetic market trajectories based on hypothetical or real-time financial news inputs. This approach allows for forward-looking risk assessment, helping to anticipate potential market movements under varying economic and geopolitical conditions.

Iterative News-Driven Market Simulation

The simulation framework operates iteratively by:

1. Generating a sequence of textual news inputs representing key macroeconomic events, earnings reports, or geopolitical developments.
2. Feeding these inputs into the Transformer-based model, which encodes the financial context and forecasts corresponding asset price movements.
3. Using the predicted time series responses as new priors to drive the next iteration of the simulation, ensuring a continuous, dynamically evolving market forecast.

By iterating over multiple simulated news sequences, the model can project alternative market paths, allowing analysts to explore the effects of both expected and unexpected events on asset prices.

Applications of Simulation in Financial Decision-Making

Simulated market responses generated by the model can be applied in several key areas:

- Risk Management: Stress-testing portfolios by introducing simulated tail-risk events (e.g., rate hikes, economic recessions, regulatory crackdowns) and evaluating their impact on asset allocations.
- Trading Strategy Evaluation: Testing algorithmic trading strategies under different simulated market conditions to ensure robustness across multiple scenarios.
- Market Impact Analysis: Quantifying how large trades or institutional moves might influence price action when combined with specific market narratives.
- Policy Scenario Forecasting: Evaluating potential macroeconomic policy effects (e.g., fiscal stimulus, Federal Reserve rate decisions) by simulating news events with varying intensities.
- Portfolio Rebalancing Sensitivity Analysis: Examining how different asset weight allocations respond to simulated external shocks, helping refine the asymmetry-aware reinforcement learning model for risk-adjusted optimization.

Monte Carlo-Driven Scenario Generation

To enhance the robustness of simulation-based insights, we extend the framework using Monte Carlo sampling, where:

$$X_t = f(X_{t-1}, \text{news}_{t-1}) + \epsilon_t$$

where:

- X_t represents the simulated asset price at time t.
- $f(X_{t-1}, \text{news}_{t-1})$ is the model's news-conditioned asset price function.
- $\epsilon_t \sim \mathcal{N}(0, \sigma)$ represents stochastic market noise, ensuring variation in simulated paths.

By running multiple Monte Carlo simulations with varying news sequences and noise realizations, we can estimate the distribution of potential market outcomes, rather than relying on a single deterministic forecast.

Practical Implementation and Use Cases

Financial institutions can leverage this simulation framework in portfolio stress testing, hedge fund strategy development, and central bank policy modeling, among other applications. Moreover, traders and asset managers can explore counterfactual market scenarios—such as the potential price impact of a hypothetical recessionary signal—by inputting custom-tailored economic narratives into the model.

Thus, simulation extends the model's utility beyond direct forecasting, offering a comprehensive tool for risk-aware financial planning and strategy refinement.

Considerations

- The low frequency model requires access to a rich corpus of earning reports, which are publicly available through EDGAR, third-party APIs.
- The high frequency model needs to be trained on several years of news articles, as well as company financial reports.
- Methods such as SHAP values or or attention mechanisms can increase the explainablity of the model and provide insights into how specific news events influence the time series predictions.
- The current asset allocation paradigm ignores transaction costs and tax implications. In practice, the costs may prove to be non-negligible especially for high frequency trades.
- The proposed asset allocation paradigm can also take into account the possibility of shorting assets. In a slightly modified format, in every sampling distribution, long one dollar on the asset with the highest return, and equivalently short the asset with the largest negative return.
- Adaptive learning can be the next stage of the model. Upon successful initial performance, we can develop mechanisms for the model to adapt to changing market conditions and news patterns. This could involve online learning techniques where the model continuously updates itself based on new data.
- Leveraging pre-trained language models (e.g., BERT, GPT) for the encoder part can enhance the model's understanding of the semantic content in news articles, leading to better performance with less training data.
- Handling vast volumes of news data, earnings reports, and macroeconomic information in real-time requires efficient storage, retrieval, and computational frameworks.
- Implementing intelligent filtering mechanisms and asynchronous processing can help manage data latency and reduce noise in AI-driven financial signals.
- AI models trained on financial news are vulnerable to misinformation, necessitating robust validation mechanisms to filter unreliable sources.
- Portfolio shifts driven by AI-generated signals must comply with risk constraints and regulatory requirements to prevent excessive exposure and unintended biases.

References

[1] Augustas Degutis and Lina Novickytė. The efficient market hypothesis: A critical review of literature and methodology. *Ekonomika*, 93(2):7–23, 2014.

[2] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Burton G Malkiel. Efficient market hypothesis. In *Finance*, pages 127–134. Springer, 1989.

[4] Lasse Heje Pedersen. *Efficiently inefficient: how smart money invests and market prices are determined*. Princeton University Press, 2019.

[5] Alexandra Gabriela Ţiţan. The efficient market hypothesis: Review of specialized literature and empirical research. *Procedia Economics and Finance*, 32:442–449, 2015.