When Words Move Markets: Incorporating a FinBERT-Derived News-Sentiment Factor into the Fama–French Framework

Alex Chow, Toto Guliyev, Michael Murphy, and Kami Seiki

Abstract

Recent advancements in natural language processing (NLP) have enabled investors to extract sentiment signals from unstructured financial text such as news headlines and earnings call transcripts. This study investigates whether incorporating a sentiment factor derived from real-time financial news can improve traditional asset pricing models. Using ticker-specific headlines sourced via Yahoo Finance, we apply a fine-tuned FinBERT sentiment analyzer to generate daily sentiment scores for S&P 500 companies. These scores are aggregated to construct a sentiment factor, which is integrated into a Long Only Momentum-based strategy. We evaluate the model's performance on historical stock return data, comparing cumulative returns with and without the sentiment factor. Our findings suggest sentiment analysis can improve a trading strategy. However, results also reveal challenges in finding a suitable model to pair with sentiment analysis.

Introduction

We construct an end-to-end pipeline that links public discourse to market behavior in order to test whether society's prevailing mood toward a firm translates into tradable price dynamics. Using the following companies: Google, Nvidia, JP Morgan, Pepsico, Adobe, Netflix, we scrape the daily news feed that underlies Yahoo Finance headlines, pass each headline through the FinBERT language model to quantify its tone on a 1 to +1 scale, and aggregate those scores into weekly sentiment values. We then overlay this sentiment series onto the stock's close-to-close returns, generating a simple trading signal that is long when the average tone is materially positive, short when it is materially negative, and flat otherwise. By tracking the cumulative equity curve and computing the strategy's Sharpe ratio, we obtain a first-pass gauge of how strongly collective sentiment, which proxies for society's real-time assessment of the company, maps into predictable excess returns and, by extension, whether incorporating such a sentiment factor might improve traditional asset-pricing models.

Data Set

News data was derived from Financial News and Stock Price Integration Dataset (Zdong104) which included 15.7 million financial news records for 4,775 S&P500 companies from 1999 to 2023. The dataset used in our methodology was filtered only include information on the following tickers from 2018 - 2020: GOOG, NVDA, JPM, PEP, ADBE, NFLX.

	Date	Article_title	Stock_symbol	Url	Publisher
0	2020-06-08 06:57:41+00:00	UBS Maintains Buy on Adobe, Raises Price Targe...	ADBE	https://www.benzinga.com/news/20/06/16202690/u...	Benzinga Newsdesk
1	2020-06-04 06:48:11+00:00	Stocks That Hit 52-Week Highs On Thursday	ADBE	https://www.benzinga.com/news/20/06/16180865/s...	Benzinga Insights
2	2020-05-27 06:24:38+00:00	Shares of several companies in the broader tec...	ADBE	https://www.benzinga.com/wiim/20/05/16116988/s...	Benzinga Newsdesk
3	2020-05-26 06:32:14+00:00	Stocks That Hit 52-Week Highs On Tuesday	ADBE	https://www.benzinga.com/news/20/05/16106805/s...	Benzinga Insights
4	2020-05-20 06:19:31+00:00	Shares of several technology companies are tra...	ADBE	https://www.benzinga.com/wiim/20/05/16075931/s...	Benzinga Newsdesk

Stock returns data was derived from yfinance's library and filtered under the same conditions

symbol	date	ADBE	GOOG	JPM	NFLX	NVDA	PEP
55	2019-01-28	-0.039862	0.112574	0.153165	0.036634	-0.134479	0.008221
56	2019-02-04	NaN	0.232349	0.014100	0.359516	-0.323633	NaN
57	2019-02-11	-0.292161	0.090810	0.234157	-0.091428	-0.230971	0.754394
58	2019-02-18	-0.806656	-0.062893	0.031114	NaN	0.199453	0.455883
59	2019-02-25	NaN	-0.182325	0.222883	-0.054033	0.819164	-0.020723

Methods

Utilizing finBERT's sentiment analysis model, we produced a sentiment value for every news headline in our dataset given our selected time frame of two years.

```python
import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification, pipeline
import pandas as pd

raw_news = list(zip(df["Article_title"], df["Date"], df["Stock_symbol"]))

sent_pipe = pipeline(
    "sentiment-analysis",
    model="ProsusAI/finbert",
    tokenizer="ProsusAI/finbert",
    top_k=None,  # return logits for all three labels
    device=0 if torch.cuda.is_available() else -1,
)
def score_headlines(news_triples):
    """
    Map each headline to a signed score:
      +1 * P(Positive)  - 1 * P(Negative)
      (Neutral contributes 0)
    """
    texts = [h for h, _, _ in news_triples]
    outs  = sent_pipe(texts, batch_size=32)
    for (headline, ts, symbol), logits in zip(news_triples, outs):
        p_pos = next(x for x in logits if x["label"] == "positive")["score"]
        p_neg = next(x for x in logits if x["label"] == "negative")["score"]
        yield ts.normalize(), symbol, p_pos - p_neg

scored_news = pd.DataFrame(score_headlines(raw_news), columns=["date", "symbol", "sent"])
scored_news.head()
```

Using this information, we created an aggregated value for each stock for every week

	date	GOOG	NVDA	JPM	PEP	ADBE	NFLX
0	2018-01-08	0.018172	0.035190	0.045384	-0.010870	0.052390	0.053526
1	2018-01-15	0.013589	0.031976	0.003018	0.023769	0.003486	-0.003480
2	2018-01-22	0.033696	0.057451	0.029290	0.013231	0.028458	0.245577
3	2018-01-29	-0.054378	-0.040316	-0.017538	-0.025378	-0.028117	-0.026111
4	2018-02-05	-0.066661	-0.006166	-0.037102	-0.063116	-0.039102	-0.067158

We then created our LOM strategy whereby each week, our strategy examines the last 5 weeks of information, calculates each assets' sharpe ratios, and

longs the top four stocks.

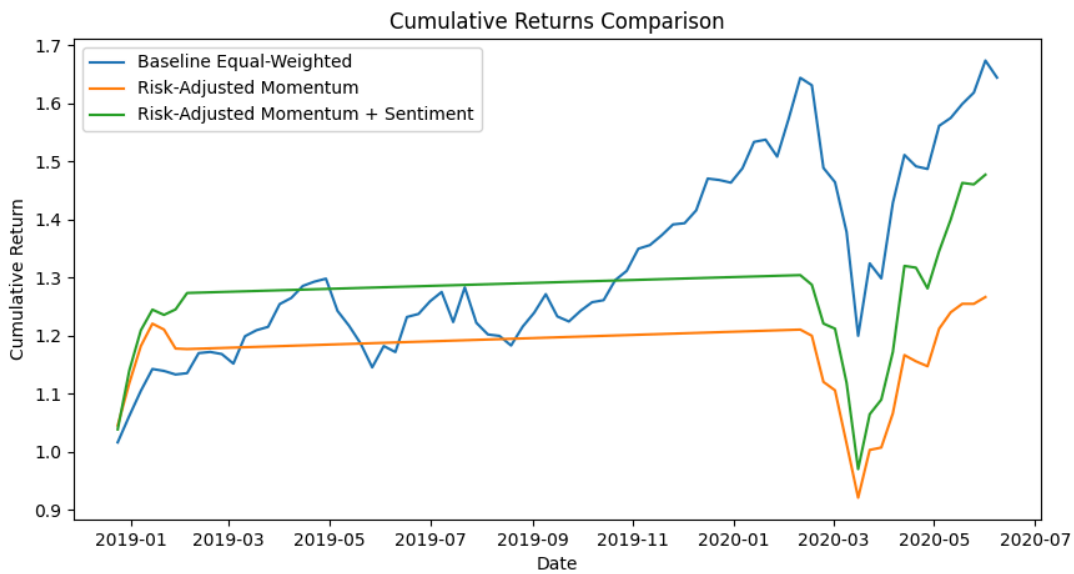
```
def compute_risk_adjusted_momentum_portfolio(returns, lookback=5):
    rolling_mean = returns.rolling(lookback).mean().shift(1)
    rolling_std = returns.rolling(lookback).std().shift(1)
    sharpe_scores = (rolling_mean / rolling_std).dropna()

    portfolio_returns = []
    for date in sharpe_scores.index:
        weekly_returns = returns.loc[date]
        weekly_sharpe = sharpe_scores.loc[date]
        ranked = weekly_sharpe.rank()
        top_mask = ranked >= ranked.quantile(0.6) # top 40% of stocks in portfolio based on sharpe
        top_return = weekly_returns[top_mask].mean()
        portfolio_returns.append(top_return)

    return pd.Series(portfolio_returns, index=sharpe_scores.index)
```

Here, we utilize the same LOM strategy in tandem with our sentiment values to long stocks with an additional, top-two sentiment, criteria.

```
def compute_momentum_sentiment_portfolio(returns, sentiment, lookback=5):
    rolling_mean = returns.rolling(lookback).mean().shift(1)
    rolling_std = returns.rolling(lookback).std().shift(1)
    sharpe_scores = (rolling_mean / rolling_std).dropna()

    portfolio_returns = []
    for date in sharpe_scores.index:
        weekly_returns = returns.loc[date]
        weekly_sentiment = sentiment.loc[date]

        weekly_returns.index = [col.replace('_Return', '') for col in weekly_returns.index]
        weekly_sharpe = sharpe_scores.loc[date]
        weekly_sharpe.index = weekly_returns.index
        weekly_sentiment.index = weekly_returns.index

        momentum_ranked = weekly_sharpe.rank()
        top_momentum_mask = momentum_ranked >= momentum_ranked.quantile(0.6) # top 40% of stocks in portfolio based on sharpe

        sentiment_ranked = weekly_sentiment[top_momentum_mask].rank()
        top_sentiment_mask = sentiment_ranked >= sentiment_ranked.quantile(0.5)

        final_mask = top_momentum_mask[top_momentum_mask].index[top_sentiment_mask]
        top_return = weekly_returns.loc[final_mask].mean()
        portfolio_returns.append(top_return)

    return pd.Series(portfolio_returns, index=sharpe_scores.index)
```

Results

Below is a graph depicting the cumulative returns for a baseline equal-weighted portfolio against a Long Only Momentum-based (LOM) strategy and a LOM strategy with sentiment analysis from December 2018 to July 2020.



From the graph, it appears that the LOM with the sentiment performs the best among the three strategies between the start date, December 2018, to November 2019, achieving the highest cumulative return of about 130% of the portfolio. After this period, the baseline equal weighted strategy seems to generally outperform both LOM models, with a max cumulative return of almost 170% around July 2020. In comparison, during the same period, the LOM with and without sentiment analysis have cumulative returns of 150% and 120% respectively. Furthermore, the minimum cumulative return was produced by the LOM without sentiment analysis, with a cumulative return of approximately 90%. Another notable observation is that, of the three strategies, the baseline strategy is the only one that did not have a cumulative return less than 100% throughout the entire period of this study.

Conclusion

The LOM with sentiment analysis has the highest cumulative return from the beginning of December 2018 to November 2019, implying that it can help a LOM strategy generate higher returns. However, during periods of greater volatility, such as during the beginning of 2020, the LOM with sentiment seems to struggle, producing portfolio decisions that lose money. This is reflected by the cumulative return of the LOM model with sentiment analysis, which was 95% during the month of March. Thus, the LOM model with sentiment analysis does have the capacity to increase portfolio returns, but primarily during periods of high unpredictability in the stock market.

Limitations

While our findings suggest that a FinBERT-derived sentiment factor can add explanatory power to the Fama–French framework, several caveats temper their generality. We hand-picked six liquid, recognisable U.S. stocks. These firms are disproportionately large-cap, media-intensive, and information-rich; their news flow, analyst coverage, and investor attention differ markedly from that of thinly traded small-caps, international issuers, or private firms. Consequently, any performance attributable to sentiment may be overstated relative to the broader equity universe.

References

Han, Z., & Fan, X. (2025, January 16). *FNSPID: A Comprehensive Financial News Dataset in Time Series*. GitHub. https://github.com/Zdong104/FNSPID_Financial_News_Dataset?tab=readme-ov-file

Kirtac, K., & Germano, G. (2024, December 26). Sentiment trading with large language models. https://arxiv.org/abs/2412.19245