# Predictability of Post-Earnings Announcement Drift with Textual and Contextual Factors of Earnings Calls

Andy Chung
Department of Advanced Interdisciplinary Studies,
Graduate School of Engineering,
The University of Tokyo
Tokyo, Japan
andy@g.ecc.u-tokyo.ac.jp

Kumiko Tanaka-Ishii
Department of Computer Science and Engineering,
School of Fundamental Science and Engineering,
Waseda University
Tokyo, Japan
kumiko@waseda.jp

## ABSTRACT

Post-Earnings Announcement Drift (PEAD), a well-known anomaly in financial markets, describes the tendency of cumulative stock returns to drift in the direction of an earnings surprise for a prolonged period following an earnings announcement. Numerous studies have used a supervised learning approach to predict PEAD, using earnings, fundamental and technical factors. However, there is a lack of study on how the context of the earnings call can be used for the PEAD prediction task. This paper uses computational linguistics techniques and large language models to examine the effectiveness of incorporating textual and contextual features from earnings calls for the PEAD prediction task. Our proposed supervised model includes four categories of features: 1) textual features, 2) contextual features, 3) earnings features, and 4) fundamental and technical features. We study the proposed model using earnings from 2010/01/01 to 2022/12/31 of all point-in-time S&P500 constituents in the US stock market. Our results show that contextual features provide information unexplained by earnings, fundamental and technical features, improving the average returns per trade of a hypothetical long-short portfolio against baseline solution in out-of-sample across all four different abnormal return calculations, ranging from 53 to 354 basis points and 16.9% to 108.5% improvement from baseline model, which uses only earnings, fundamental and technical features.

## CCS CONCEPTS

• **Applied computing** → **Economics**.

## KEYWORDS

Post-earnings announcement drift, earnings call, computational linguistics, large language models, machine learning

**ACM Reference Format:**
Andy Chung and Kumiko Tanaka-Ishii. 2023. Predictability of Post-Earnings Announcement Drift with Textual and Contextual Factors of Earnings Calls. In *4th ACM International Conference on AI in Finance (ICAIF '23), November 27–29, 2023, Brooklyn, NY, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3604237.3626861



*ICAIF '23, November 27–29, 2023, Brooklyn, NY, USA*

ACM ISBN 979-8-4007-0240-2/23/11.
https://doi.org/10.1145/3604237.3626861

## 1 INTRODUCTION

The post-earnings announcement drift (PEAD), a well-known market anomaly [3], is the tendency for a stock price to drift toward an earnings surprise for a period after an earnings announcement. Inconsistent with the Efficient Market Hypothesis [13] states, the post-earnings announcement drift associated with earnings surprise does not lead to an immediate adjustment of the stock price but instead to a predictable drift of the stock price, which could last 60 trading days after the earnings announcement [16].

Previous studies have found that conference calls provide information to market participants in addition to the information in the corresponding press release and company fillings [17]. However, previous studies related to post-earnings announcement drift modeling do not fully make use of earnings call information, for example, only to the extent of sentiment analysis of the entire corpus using the sentiment lexicon [37] [43] [44].

This paper proposes a systematic approach to derive textual and contextual features from earnings calls, including a text preprocessing procedure, ChatGPT for abstractive summarization, and a sentence-transformer-based contextual embedding extraction, followed by modeling PEAD with machine learning models. The main purpose of this paper is to examine the effectiveness of incorporating earnings calls into the PEAD prediction task.

Our primary contributions are the following:

(1) We propose the earnings call factor, which consists of textual and contextual features derived from the earnings call, for PEAD prediction.
(2) Our experiment shows that contextual features from earnings calls provide information unexplained by earnings, fundamental and technical features, improving models' ability to predict PEAD.

## 2 RELATED WORK

### 2.1 Modeling post-earnings announcement drift

PEAD is one of the most robust and well-studied anomalies in the asset pricing literature. [22] shows evidence that transient institutional investors actively trade and profit from the PEAD anomaly. [23] propose adding a risk factor related to unexpected earnings surprise in Fama and French's three-factor risk model [14]. They show that the earnings surprise risk factor provides an improvement in explaining post-earnings announcement drift when included in addition to the three factors of Fama and French. After adjusting raw returns for the four risk factors, cumulative abnormal returns over the 60 trading days after quarterly earnings announcements

are economically and statistically insignificant [23]. [37] proposes using the random forest to predict PEAD, using valuation ratios, forecast errors, forecast uncertainty, and the sentiment of the earnings call. Similarly, [43] [44] also demonstrate the promising result by applying various machine learning models on traditional factors for the PEAD prediction task. Although their works consider extracting the earnings call feature as the input, only the sentiment feature is used.

## 2.2 Lexicon-based Sentiment analysis

Sentiment Analysis (SA) or Opinion Mining (OM) is the computational study of people's opinions, attitudes, and emotions toward an entity, individuals, issues, events, topics, and their attributes [30]. A lexicon-based approach for sentiment analysis is to manually select keywords to form a lexicon dictionary guided by a pre-defined methodology. The Harvard Psychosociological Dictionary, specifically the Harvard-IV-4 TagNeg (H4N) file [1], is a list of positive and negative words using the manual approach. Due to polysemy, a domain-specific lexicon dictionary often performs better in a specific domain, such as finance. [28] shows that in a large sample of 10-K fillings from 1994 to 2008, almost three-fourths of the words identified as negative by the widely used Harvard Dictionary are words typically not considered negative in financial contexts. They developed a dictionary that better reflects the tone in financial context by linking word lists with 10-K filings returns, trading volume, return volatility, fraud, material weakness, and unexpected earnings [28].

## 2.3 Text readability

Assessing text readability has a long history and is particularly useful for educational purposes, such as selecting literary passages for different grades or grading student writing [8] [9]. As a result, prior work on readability focuses mainly on labeling texts with the appropriate school grade level. The Flesch Reading Ease (FRES) [24] works by counting the number of words, syllables, and sentences in the text. The Flesch-Kincaid grade level is widely used in education and is similar to FRES but with different weighting factors. Gunning fog index [19] uses the number of complex words (based on syllables), the total number of words and sentences. SMOG grade [27] uses the total number of polysyllables relative to the total number of sentences. The automatic readability index (ARI) [38] uses the total number of characters, words, and sentences. The Coleman-Liau index [7] uses the average number of letters per 100 words and the average number of sentences per 100 words. Linsear Write [25] uses the average total number of words with two syllables or less per 100 words and the average total number of words with three syllables or more per 100 words. Dale–Chall readability [10] uses the number of complex words, the total number of words and sentences.

## 2.4 Context analysis and contextual embeddings

Communication in natural languages depends on the use of contextual information to elaborate what has been said literally, to eliminate ambiguity, and to further specify the content [18]. While how to represent the context and utilize contextual information remains a challenging question, recent advancement in large language models provides a new way to represent context from the text as a form of numerical vector.

Semantic representation and measuring semantic similarity between text data is one of the important research problems in NLP [6]. Contextual embeddings provide vector representations of words or sentences such that semantically similar words or sentences are closed in vector space. Word2vec [31] is a neural network that learns word associations from a large corpus of text. Word2vec embeddings capture the semantic and syntactic qualities of words, and, as a result, using cosine similarity can indicate the level of semantic similarity between words. [1] proposes attention, a mechanism to learn and assign weights based on the relevance of words while training a language model. Attention becomes the foundation of transformers [41] and BERT [12] and fine-tuning has become an essential technique to adopt the pre-trained language model to another domain [12]. Although pre-trained BERT shows human-level performance on a wide range of language understanding tasks [21], it is not suitable for semantic textual similarity tasks. The most commonly used approach is to pool BERT embedding as contextual representation, however, [35] provide evidence that BERT embedding is worse than an average GloVe embedding [32] for the semantic textual similarity task. Sentence-BERT [12] addresses the issue by modifying the pre-trained BERT network with the structure of the siamese and triplet networks, allowing the model to generate embeddings that can be compared using cosine similarity or Euclidean distance.

While there is solid evidence that pre-trained language models fine-tuned on large and diverse supervised datasets improve performance on the out-of-domain task, a domain with a lack of clean textual data and proper labels is challenging to the fine-tuning task. [33] shows that fine-tuning the target task may not perform better than simple feature extraction from a pre-trained model if the pre-training and target tasks are very different domains. [36] demonstrated that a feature extraction approach without adaptations to the target domain performs very well and suppresses other methods in a legal case entailment task. Given limited labeled data in a particular domain, models with little or no adaptation to the target task can be more robust than fine-tuning models. In our study, the model weights are frozen and the pre-trained representations are used for downstream models.

# 3 METHODOLOGY

## 3.1 Post-earnings stock return and abnormal return calculation

We propose using four different post-earnings stock return calculations to derive the target variable and for performance evaluation:

(1) Post-earnings buy-and-hold stock return ($BnH$)
(2) Post-earnings buy-and-hold stock return with index return hedging ($BnH_{hedged}$)
(3) Post-earnings buy-and-hold abnormal return ($BnH_{abnormal}$)
(4) Post-earnings cumulative abnormal return ($CAR$)

[1] http://www.wjh.harvard.edu/~inquirer/

*3.1.1 Post-earnings buy-and-hold stock return (BnH).* Since earnings announcements are scheduled in pre-market open or after-market close, the starting price after earnings differs for the two cases to ensure no look-ahead bias in the study. In the first case, the return is calculated from the open price of the earnings announcement date, whereas in the latter case, the return is calculated starting from the open price of the next trading day after an earnings announcement. We assume that an investor opens a position and holds the position for the next 60 trading days and closes at the closing price at the 60-th trading days [16].

The return of stock $i$ following an earnings event starting on the trading day $t$ with a holding period of $T$ days is defined as

$$r_{i,t,T} = \frac{p_{\text{close},i,t+T-1}}{p_{\text{open},i,t}} - 1, \tag{1}$$

where $p_{\text{open},i,t}$ denotes the opening price of stock $i$ on day $t$ and $p_{\text{close},i,t+T-1}$ denotes the closing price of stock $i$ on day $t+T-1$. Both prices are adjusted for dividends, stock splits, and any other corporate actions. The post-earnings buy-and-hold stock return (BnH) on trading day $t$ of stock $i$ with 60 days holding period is thus

$$BnH = r_{i,t,60}. \tag{2}$$

*3.1.2 Post-earnings buy-and-hold stock return with index return hedging ($BnH_{hedged}$).* In our study, the stocks we considered are S&P500 index constituents. Therefore, the index used for hedging is the S&P500 index.

The index return on trading day $t$ with a holding period of $T$ days is defined as

$$R_{t,T} = \frac{P_{\text{close},t+T-1}}{P_{\text{open},t}} - 1, \tag{3}$$

where $P_{\text{open, t}}$ denotes the opening price of the index on day $t$ and $P_{\text{close, t+T-1}}$ denotes the closing price of index on day $t+T-1$. The post-earnings buy-and-hold stock return with index return hedging on the trading day $t$ of stock $i$ with a 60-day holding period is

$$BnH_{hedged} = r_{i,t,60} - \beta_i \cdot R_{t,60}, \tag{4}$$

where $\beta_i$ is the beta of the stock $i$ to the index, calculated based on the historical return of the past 60 days.

*3.1.3 Post-earnings buy-and-hold abnormal return.* In line with previous studies [29] [2] [4], the buy-and-hold abnormal return is defined as the next 60 days' stock return after an earning event, minus the expected return during the same period.

$$BnH_{abnormal} = r_{i,t,60} - E(r_{i,t,60}) \tag{5}$$

where $E(r_{i,t,60})$ denotes the expected return of stock $i$ with a 60 days period after earnings. In our study, the stock return of a size-matched controlled firm in the same sector from S&P500 constituents is used as an estimate of the expected return $E(r_{i,t,60})$ [29] [37].

*3.1.4 Post-earnings cumulative abnormal return.* A related, slightly different version of the buy-and-hold abnormal returns is the cumulative abnormal return (CAR) [29]. While the buy-and-hold abnormal returns answer whether firms earned abnormal stock returns over a particular analysis horizon, the CAR answers whether firms persistently earn abnormal monthly returns [29]. The cumulative abnormal return of the stock $i$ over the next 60 trading days is defined as

$$CAR = \sum_{s=t}^{t+59} (DR_{i,s} - DR_{j,s}) \tag{6}$$

where stock $j$ is the size-matched, same sector controlled firm of stock $i$ and $DR_{i,s}$, $DR_{j,s}$ are the simple daily return of stock $i$ and $j$ from day $s$ open to day $s+1$ open, respectively.

## 3.2 Target variable

We consider a binary target derived from the post-earnings buy-and-hold stock return ($BnH$) as the classification model target. The evaluation of the model is based on four post-earnings stock return calculations, as discussed in the previous section.

Below are five possible scenarios for the stock $i$ following the earnings announcement on day $t$:

(1) Estimated earnings equal to actual earnings result;
(2) $BnH > 0$ and beat earnings (estimated earnings is lower than actual earnings result);
(3) $BnH \leq 0$ and beat earnings;
(4) $BnH \geq 0$ and miss earnings (estimated earnings is higher than actual earnings result);
(5) $BnH < 0$ and miss earnings.

Scenario (1) is a situation in which the market already digests all available information and fully aligns with the actual result. Therefore, this scenario is not of our interest and is filtered out in our study. The study focuses on scenarios (2) to (5).

Scenario (2) and scenario (5) indicate that both the direction of post-earnings stock returns and earnings surprises are the same. We label these two scenarios with the target $y = 1$ (Is PEAD). On the contrary, scenarios (3) and (4) indicate that both the direction of post-earnings stock returns and earnings surprises are the opposite. We label these two scenarios with the target $y = 0$ (not PEAD).

Based on this, we are interested in developing a classification model $f$ based on the feature set $X_{i,t}$ on day $t$ of the stock $i$ to predict the probability of $y = 1$ (Is PEAD):

$$f(X_{i,t}) = \hat{y} = \hat{\mathbb{P}}(y = 1). \tag{7}$$

## 3.3 Earnings call transcript pre-processing and abstractive summarization

An earnings call transcript, denoted as $T$, is a text corpus recording all conversations on a company earnings call. The transcript is divided into the MD&A and Q&A sessions, denoted as $T_a$ and $T_b$. In some cases, the company may have decided not to hold the Q&A session.

$T_a$ and $T_b$ can be further segmented by speakers. Denote the total number of distinct executive speakers on MD&A as $n$. The text in

MD&A session can be segmented and concatenated with the corresponding speaker (CEO, CFO, CTO, etc.), denoted as $\{T_{a,1}, \ldots, T_{a,n}\}$. Similarly, denote the total number of distinct analysts asking questions as $m$. The text in Q&A session can be further segmented and concatenated with the corresponding analysts and the corresponding answers from the company, denoted as $\{T_{b,1}, \ldots, T_{b,m}\}$.

We propose using text summarization as a preprocessing step for the corpus. Text summarization potentially benefits downstream tasks in textual features and contextual embedding extraction. In particular, reducing text length is important for the BERT model due to the quadratic complexity corresponding to the sequence length. There are two main approaches for text summarization: extraction summarization and abstractive summarization [20]. Extraction summarization concatenates salient text units from the original corpus, whereas abstractive summarization generates novel sentences that summarize information from the corpus. Our proposed approach uses ChatGPT [2] for abstractive summarization. ChatGPT is a Large Language Model developed recently by OpenAI, trained on the OpenAI 175B parameter foundation model and a large corpus of text data from the Internet via reinforcement and supervised learning methods. Text summarization is performed by adding "/n/nTl;dr" to the end of the original text. Below is an example of summarization done by ChatGPT. It is worth noting that ChatGPT summarization is non-deterministic, and the example is just one possible summarization generated by ChatGPT.

*Prompt:*

> **Analyst**: Hi, everyone. Thanks for taking my question. So you recently adjusted prices, and that may have put many of your competitors in the back foot. In addition to that, capital markets have recently gotten a lot tougher. So with those factors in mind, I'm curious how you see the current competitive landscape changing over the next few years. And who do you see as your chief competitors five years from now?
> **Elon Musk**: Five years is a long time. As with the Tesla order part, AI team, until late last night, and just we're asking [Inaudible], so who do we think is close to Tesla with – a general solution for self-driving? And we still don't even know really who would even be a distant second. So yes, it really seems like we're – I mean, right now, I don't think you could see a second place with a telescope, at least we can't. So that won't last forever. So in five years, I don't know, probably somebody has figured it out. I don't think it's any of the car companies that we're aware of. But I'm just guessing that someone might figure it out eventually. So yes.
>
> Tl;dr

*Summarization by ChatGPT:.*

> Elon Musk doesn't see any current competitors to Tesla's self-driving technology, and can't predict who their chief competitors will be in five years. He thinks someone may eventually figure it out, but it's not currently any of the car companies they're aware of.

[2]https://openai.com/blog/chatgpt

Using $T_a$ and $T_b$, the summarization generated by ChatGPT is denoted as $S_a$ and $S_b$ respectively.

## 3.4 Feature extraction

Our proposed model includes four categories of features:

(1) Textual features;
(2) Contextual features;
(3) Earnings features;
(4) Fundamental and technical (FnT) features.

These features are used for the downstream supervised model to predict the target we define in Section 3.2. All the features are listed in Table 1. The median of the feature in the training set replaces the corresponding feature with a missing value.

For textual features and contextual features, features can be generated using 1) the original transcript $T_a$, $T_b$, $\{T_{a,1}, \ldots, T_{a,n}\}$, $\{T_{b,1}, \ldots, T_{b,m}\}$ or 2) the summarized text generated by ChatGPT $S_a$ and $S_b$. We denote the two different text preprocessing approaches as 1) Original and 2) ChatGPT, respectively. Given that the pretrained language model has an input token length limit, any exceeding token would be discarded. If the input is a list of text, such as $\{T_{a,1}, \ldots, T_{a,n}\}$, the feature or the prediction is generated by averaging all the questions and answer pairs.

*3.4.1 Textual features.* Textual features compute textual characteristics and n-gram statistics from the earnings call transcript.

*Sentiment features.* Two lexicon-based dictionaries are used for sentiment score extraction, namely Harvard IV-4 and Loughran & McDonald's sentiment dictionaries [28]. They are sentiment dictionaries for general and financial corpus respectively. Similar to [5], the aggregated sentiment score is obtained for each corpus. Positive and negative word occurrences are obtained in the corpus using the sentiment lexicon. The sum of positive and negative sentiment scores ($sp$, $sn$) within that corpus and the total number of positive and negative sentiment words ($c$) are calculated. Polarity [5] of the corpus is calculated as

$$Polarity = \frac{sp}{sp + sn} \tag{8}$$

and the subjectivity [5] of the corpus is calculated as

$$Subjectivity = \frac{sp + sn}{c} \tag{9}$$

*Readability features.* The following readability metrics are used for readability features:

- The Flesch Reading Ease [24]
- Flesch–Kincaid Grade Level [24]
- Gunning fog index [19]
- SMOG grade [27]
- Automated readability index [38]
- Coleman–Liau index [7]
- Linsear Write [25]
- Dale–Chall readability [10]

**Table 1: Overview of all features considered in this study.**

| Feature ID | Feature type | Feature sub-type | Feature name and description | Hyper-parameters |
|---|---|---|---|---|
| 1 | Textual | Sentiment | Harvard IV-4 Sentiment | - |
| 2 | | | Loughran & McDonald's Sentiment | - |
| 3 | Textual | Readability | Flesch Reading Ease | - |
| 4 | | | Flesch–Kincaid Grade Level | - |
| 5 | | | Gunning Fog Index | - |
| 6 | | | SMOG Grade | - |
| 7 | | | Automated Readability Index | - |
| 8 | | | Coleman–Liau Index | - |
| 9 | | | Linsear Write | - |
| 10 | | | Dale–Chall Readability | - |
| 11 | Contextual | - | Sentence-BERT embeddings | - |
| 12 | Earnings | - | Earnings per share $EPS(t)$ | $t \in \{0, 1, 2, 3, 4\}$ |
| 13 | | | Consensus estimate earnings per share $\hat{EPS}(t)$ | $t \in \{0, 1, 2, 3, 4\}$ |
| 14 | | | Earnings surprise $\Delta EPS(t)$ | $t \in \{0, 1, 2, 3, 4\}$ |
| 15 | | | Post-earnings open gap on stock/excess return | - |
| 16 | FnT | Fundamental | Current Ratio - CR($q$) | $q \in \{0, 1, 2, 3, 4\}$ |
| 17 | | | P/E Ratio - PE($q$) | |
| 18 | | | P/B Ratio - PB($q$) | |
| 19 | | | P/S Ratio - PS($q$) | |
| 20 | | | Debt-to-Equity Ratio - DE($q$) | |
| 21 | | | Free Cash Flow - FCF($q$) | |
| 22 | | | Debt-to-Equity Ratio - DE($q$) | |
| 23 | | | Dividend Yield Ratio - DY($q$) | |
| 24 | FnT | Technical | EWMA($\tau$) of historical stock/excess return | $\tau \in \{5, 10, 20, 40, 60\}$ |
| 25 | | | Volatility($t$) of historical stock/excess return | $t \in \{20, 40, 60\}$ |

*3.4.2 Contextual features.* Unlike textual features, which are contextually independent, contextual features capture the meaning of the entire corpus. Contextual embeddings assign corpus a numerical vector based on its context. We use sentence-BERT, a modification of BERT that uses siamese and triplet networks. These embeddings are semantically meaningful and can be compared on the basis of distance, making them suitable for use as a feature for downstream supervised learning. The pre-trained language model we used is ALL-MPNET-BASE-V2 [3], a sentence-transformers model fine-tuned on a 1B sentence pairs dataset based on the MPNet model [39].

*3.4.3 Earnings features.*

*Earnings surprises features.* We define the earnings surprise as the difference between the earnings announcement and the consensus earnings forecast, normalized by the share price, in line with prior work [11] [26].

The consensus estimate earnings are the average of all analysts' estimates, denoted as $\hat{EPS}$. Based on $\hat{EPS}$ and the actual earnings per share ($EPS$), the earnings surprise ($\Delta EPS$) is calculated as

$$\Delta EPS = \frac{EPS - \hat{EPS}}{P_{\text{open},i,t}}. \tag{10}$$

The earnings surprise feature includes a hyperparameter $q \geq 0$, denoted as $\Delta EPS(q)$, indicating the quarter lag of the earnings surprise. For example, $\Delta EPS(0)$ and $\Delta EPS(1)$ are the latest earnings surprise and previous earnings surprise, respectively.

In addition, the opening gap on the first day after the earnings announcement is used based on stock and excess returns. The excess return is the stock return minus the index return during the same period. The opening gap is the stock return from yesterday's closing price to today's opening price.

The following are the earnings features we used:

- Earnings per share $EPS(t)$
- Consensus estimate earnings per share $\hat{EPS}(t)$
- Earnings surprise $\Delta EPS(t)$
- Post-earnings opening gap on stock/excess return

*3.4.4 Fundamental and technical (FnT) features.*

*Fundamental features.* Fundamental features are various financial ratios based on the financial report. The following are the fundamental features that we used:

- Current Ratio (CR) is calculated as the total current assets divided by the total current liabilities.
- P/E ratio (PE) is the ratio of the stock's current price to earnings per share.
- P/B ratio (PB) is the ratio of the stock's current price to the book value of equity.
- P/S ratio (PS) is the current price of the stock divided by the company's revenue per share.

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

- Debt-to-Equity ratio (DE) is the ratio of the company's debt to the value of total shareholder equity.
- Dividend yield ratio (DY) is the dividend per share divided by the price per share.

Similarly to the earnings features, all the fundamental features are associated with the hyperparameter $q \geq 0$, indicating the quarter lag of the generated feature, with $q = 0$ representing numbers extracted from the latest available company report.

*Technical features.* Technical features are various factors based on the price and volume of the stock and index. The following are the technical features that we used:

- Exponentially Weighted Moving Average (EWMA) with half-life $\tau$ on both historical stock return and excess return.
- Volatility on both historical stock return and excess return. The calculation is $\sigma(t) \cdot \sqrt{t}$, where $t$ is the lookback window and $\sigma(t)$ is the standard deviation of return on $t$ lookback window.

## 3.5 Model

Our features consist of numerical features and embedding features, and they are further classified into four different feature types: earnings, FnT, textual, and contextual. Since our study aims to understand whether a value is added to the prediction using textual and contextual features, we propose using *blending* as an extra step to generate four aggregated linear factors on the training data, using the corresponding features grouped by feature type. Finally, a logistics regression is fitted on the validation data using only the four aggregated linear factors (blended features) based on the target we defined in Section 3.2.

*3.5.1 Blending.* Stacked generalization is a general method of using a high-level model to combine lower-level models to achieve greater prediction accuracy [42] [40]. The idea of stacking is to generate an out-of-sample prediction on the original data by partitioning the training data into $N$ partition. Each partition is considered an out-of-sample for the rest of the $N-1$ partition. A stacked feature is the prediction on a single partition generated by a lower-level model fitted on the rest of the $N-1$ partition. Finally, a high-level model is fitted based on the stacked features and used as the final prediction for the testing data. A similar method, called blending, is typically used if the data have a time factor. The idea of blending is to split the data into two partitions, usually in chronological order. A low-level model fits the first partition data, and the prediction on the second partition generated by the model is called a blended feature. Like stacking, an additional high-level model is fitted to the blended features and used as the final prediction for testing data.

We use blending where the training set is considered the first partition, and the validation set is the second partition. Numerical features, which refer to textual, earnings, and FnT features, are fitted by catboost [34]. On the other hand, the embedding features, which refer to contextual features, are fitted by Linear discriminant analysis (LDA) [15]. The optimization task is classification with logloss as the objective function, based on the target we defined in Section 3.2. We used catboost for numerical feature because gradient boosting trees is empirically robust under multicollinearity and high dimensional data, and it is good at learning complex nonlinear features. On the other hand, LDA is more suitable for handling embeddings as it projects embedding features to a lower-dimensional space, such that the projected classes are both far away and have small within-group variances. The hyperparameters of both models are found by cross-validation. Finally, a logistics regression is fitted to the validation data using only the blended features:

$$f(X_{i,t}) = b_0 + b_1 x_{i,t,1} + b_2 x_{i,t,2} + b_3 x_{i,t,3} + b_4 x_{i,t,4}, \quad (11)$$

where $x_{i,t,1}, x_{i,t,2}, x_{i,t,3}$ and $x_{i,t,4}$ are blended features of earnings, FnT, textual, contextual features of stock $i$ at time $t$ respectively, and $b_1, b_2, b_3$ and $b_4$ are the corresponding coefficients of the regression model with an intercept $b_0$.

# 4 EXPERIMENT RESULTS AND DISCUSSION

Our experiment is based on the earnings from 2010/01/01 to 2022/12/31 of the point-in-time S&P500 constituents in the US stock market. Since the return is calculated post-earnings for 60 days, the data ends at the beginning of April 2023 to calculate the target for earnings in December 2022. Our training & validation, gap, and testing data split is shown in Table 2. Only earnings with a difference between the consensus estimate and actual earnings are considered in the study. The gap data split ensures that the training, validation, and test data are completely separated. Missing value imputation using its median only considers data from training data. All experimental results are reported based on the testing set.

**Table 2: Summary of train/valid/test split of data and total number of earnings.**

| Data split | Period (inclusive) | # Earnings |
|---|---|---|
| Training | 2010-01-01 to 2014-09-30 | 6275 |
| Gap (Not Used) | 2014-10-01 to 2014-12-31 | - |
| Validation | 2015-01-01 to 2019-09-30 | 7787 |
| Gap (Not Used) | 2019-10-01 to 2019-12-31 | - |
| Testing | 2020-01-01 to 2022-12-31 | 5657 |

## 4.1 Performance evaluation

We report two types of evaluations:

(1) Area Under The Curve (AUC)
(2) Long-Short portfolio returns

Our model is built based on a logistic regression using the binary target derived from *BnH*, as defined in Section 3.2. An AUC above 0.5 means the model is better than random guessing in classifying the sign of post-earnings 60-day buy-and-hold stock return. We can construct a hypothetical portfolio based on the model prediction $\hat{y}$. The higher the prediction score $\hat{y}$, the model expects a higher likelihood that the stock return follows the earnings surprise direction post-earnings announcement (PEAD), and vice versa. Thus, we consider a unit dollar long or short investments based on four scenarios:

(1) Long if the prediction score $> c_1$ and beat earnings
(2) Long if the prediction score $< c_2$ and miss earnings
(3) Short if the prediction score $> c_3$ and miss earnings

**Table 3: Model performance evaluation in testing data. $PnL_1$, $PnL_2$, $PnL_3$ and $PnL_4$ show two information separated by row: 1) portfolio returns with percentage as unit and 2) $p$-value of t-statistics in the bracket. Levels of significance: $^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.025$, $^{****}p < 0.01$.**

| Model | Text preprocessing | Selected features | AUC | $PnL_1$ | $PnL_2$ | $PnL_3$ | $PnL_4$ |
|---|---|---|---|---|---|---|---|
| Baseline | - | Earnings + FnT (Baseline) | 0.546 | 2.544 | 2.193 | 3.803 | 3.501 |
| | | | | (0.269) | (0.265) | (0.179) | (0.186) |
| Proposed | Original | Baseline + Textual | 0.529 | 0.965 | 0.797 | 2.561 | 2.231 |
| | | | | (0.406) | (0.408) | (0.262) | (0.281) |
| | | Baseline + Contextual | **0.550** | 4.698 | 2.919 | 4.444 | 5.244 |
| | | | | (0.124) | (0.196) | (0.120) | ($0.079^{*}$) |
| | | Baseline + Textual + Contextual | 0.534 | 4.930 | **4.368** | 5.125 | 4.900 |
| | | | | (0.127) | **(0.115)** | ($0.091^{*}$) | ($0.099^{*}$) |
| | ChatGPT | Baseline + Textual | 0.545 | 2.377 | 1.926 | 2.590 | 2.135 |
| | | | | (0.281) | (0.287) | (0.264) | (0.293) |
| | | Baseline + Contextual | 0.549 | 4.597 | 2.725 | 6.727 | 6.271 |
| | | | | (0.132) | (0.223) | ($0.042^{**}$) | ($0.047^{**}$) |
| | | Baseline + Textual + Contextual | 0.548 | **5.303** | 3.652 | **7.340** | **6.720** |
| | | | | **($0.098^{*}$)** | (0.149) | **($0.029^{**}$)** | **($0.034^{**}$)** |

**Table 4: Model diagnosis of logistics regression models of 1) baseline based on two blended features and intercept, and 2) proposed model with different text preprocessing based on four blended features and intercept. Levels of significance: $^{*}p < 0.1$, $^{**}p < 0.05$, $^{***}p < 0.025$, $^{****}p < 0.01$.**

| Model (Text preprocessing) | Variable | Coeff. | $t$-stats | $p$-value | VIF |
|---|---|---|---|---|---|
| Baseline | Constant | 0.4038 | 13.841 | $0.000^{****}$ | 1.583 |
| | Earnings | 0.1823 | 7.227 | $0.000^{****}$ | 1.009 |
| | FnT | 0.1195 | 2.763 | $0.006^{****}$ | 1.009 |
| Proposed (Original) | Constant | 0.3898 | 13.048 | $0.000^{****}$ | 1.667 |
| | Earnings | 0.1750 | 6.904 | $0.000^{****}$ | 1.019 |
| | FnT | 0.1081 | 2.484 | $0.013^{***}$ | 1.019 |
| | Textual | 0.0958 | 2.265 | $0.024^{***}$ | 1.017 |
| | Contextual | 0.0456 | 1.763 | $0.078^{*}$ | 1.026 |
| Proposed (ChatGPT) | Constant | 0.3923 | 13.344 | $0.000^{****}$ | 1.610 |
| | Earnings | 0.1771 | 7.001 | $0.000^{****}$ | 1.014 |
| | FnT | 0.1017 | 2.334 | $0.020^{***}$ | 1.023 |
| | Textual | 0.0104 | 0.255 | 0.799 | 1.004 |
| | Contextual | 0.0811 | 3.545 | $0.000^{****}$ | 1.020 |

(4) Short if the prediction score $< c_4$ and beat earnings

For the purpose of validating the prediction quality, the prediction threshold $c_1, c_2, c_3, c_4$ is chosen such that the above scenarios cover an equal number of cases (28), with 56 long and 56 short, which represent in total 2% of all earnings in the testing data. The choice is based on the fact that S&P500 companies are well covered by analysts, a significant earnings surprise and PEAD is expected to be rare.

Based on the long-short portfolio, using all return calculations $BnH$, $BnH_{hedged}$, $BnH_{abnormal}$ and $CAR$ defined in Section 3.1, the portfolio return can be obtained, denoted as $PnL_1$, $PnL_2$, $PnL_3$ and $PnL_4$. We report the portfolio return and the $t$-test between the long- and short-stock components. We use a one-side $t$-test based on the corresponding return calculations:

$H_0$: Returns from long- and short-stocks are the same.

$H_1$: Returns from the long stocks are greater than the short stocks

## 4.2 Results and discussion

Table 3 shows the evaluation results of the baseline and proposed models. Table 4 shows the model diagnosis of the logistics model using the four blended features and an intercept. The baseline model uses only earnings and FnT features. Our proposed model uses either the original transcripts or a summarized transcript generated by ChatGPT, followed by using 1) textual features, 2) contextual features, or 3) both textual and contextual features, in addition to the baseline model.

We found strong evidence that contextual features improve the baseline. From Table 4, the contextual feature has a $t$-statistics of 1.763 if we use the original transcript. The $t$ statistics improve significantly to 3.545 if we use the ChatGPT summarized transcript. There are two possible reasons: First, the original transcript has a longer text than the sentence-BERT maximum token limit. Second, the summarized content from ChatGPT provides normalization across all the MD&A and Q&A text, improving the performance of the downstream sentence-BERT and LDA models. The out-of-sample evaluations in Table 3 show a consistent result with the in-sample model diagnosis - all the best models across five metrics have contextual features. Models with contextual features show an improvement across all portfolio returns, ranging from 53 to 354 basis points and from 16.9% to 108.5% improvement from baseline.

Textual features, on the other hand, have mixed results. The textual features generated from the original transcripts have high $t$ statistics (2.265), while the ChatGPT version gives very low $t$ statistics (0.225). This is because ChatGPT normalizes the text and, thus, the readability of the text and possibly the sentiment of the original text, which are the source of information for the textual features. As opposed to contextual features, we recommended using the

original transcript for textual features instead of the ChatGPT summarized one. However, while the textual feature shows promising $t$-statistics using the original transcript, the out-of-sample performance shows that adding textual features alone without contextual features to the baseline is harmful. This suggests that while textual features have predictability in training and validation data, they have very different behavior in testing data. This is possibly due to alpha decay or because our experiment testing period is during the COVID-19 pandemic, in which many factors have different behaviors corresponding to stock return.

## 5 CONCLUSION

In this paper, we proposed the earnings call factor, which consists of textual and contextual features derived from the earnings call, for PEAD prediction. A systematic approach is introduced to derive textual and contextual features from earnings calls based on computational linguistics techniques and large language models. Although textual features have been useful in the past (pre-2020) for PEAD prediction, we found that they hurt the performance in out-of-sample data (post-2020). On the other hand, contextual features generated by the sentence-BERT model consistently improve PEAD prediction performance over all periods, including out-of-sample. Performance is further enhanced by using ChatGPT to generate an abstractive summary of the earnings transcript before the embedding extraction step. Based on the model diagnosis, we found that the information from the contextual features of the earnings call is unexplained by the earnings, fundamental, and technical features.

## REFERENCES


[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. https://doi.org/10.48550/ARXIV.1409.0473

[2] Karthik Balakrishnan, Eli Bartov, and Lucile Faurel. 2010. Post loss/profit announcement drift. *Journal of Accounting and Economics* 50, 1 (2010), 20–41.

[3] Ray Ball and Philip Brown. 1968. An Empirical Evaluation of Accounting Income Numbers. *Journal of Accounting Research* 6, 2 (1968), 159–178.

[4] Robert Battalio and Richard Mendenhall. 2011. Post-Earnings Announcement Drift: Bounds on Profitability for the Marginal Investor. *Financial Review* 46 (11 2011), 513 – 539. https://doi.org/10.1111/j.1540-6288.2011.00310.x

[5] Mikhail Bautin, Lohit Vijayarenu, and Steven Skiena. 2008. International sentiment analysis for news and blogs. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 2. 19–26.

[6] Dhivya Chandrasekaran and Vijay Mago. 2021. Evolution of Semantic Similarity—A Survey. *ACM Comput. Surv.* 54, 2, Article 41 (feb 2021), 37 pages.

[7] Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology* 60 (1975), 283–284.

[8] Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *ITL - International Journal of Applied Linguistics* 165 (12 2014), 97–135. https://doi.org/10.1075/itl.165.2.01col

[9] Scott A Crossley, Aron Heintz, Joon Choi, Jordan Batchelor, Mehrnoush Karimi, and Agnes Malatinszky. 2021. The CommonLit Ease of Readability (CLEAR) Corpus.. In *EDM*.

[10] Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin* (1948), 37–54.

[11] Stefano DellaVigna and Joshua M. Pollet. 2009. Investor Inattention and Friday Earnings Announcements. *The Journal of Finance* 64, 2 (2009), 709–749.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[13] Eugene F. Fama. 1970. Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance* 25, 2 (1970), 383–417.

[14] Eugene F. Fama and Kenneth R. French. 1993. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics* 33, 1 (1993), 3–56.

[15] Ronald Fisher. 1936. Linear discriminant analysis. *Ann. Eugenics* 7 (1936), 179.

[16] George Foster et al. 1984. Earnings Releases, Anomalies, and the Behavior of Security Returns. *The Accounting Review* 59, 4 (1984), 574–603.

[17] Richard Frankel, Marilyn Johnson, and Douglas J. Skinner. 1999. An Empirical Examination of Conference Calls as a Voluntary Disclosure Medium. *Journal of Accounting Research* 37, 1 (1999), 133–150. http://www.jstor.org/stable/2491400

[18] Barbara J. Grosz. 1995. Essential Ambiguity: The Role of Context in Natural-language Processing.

[19] Robert Gunning. 1969. The Fog Index After Twenty Years. *Journal of Business Communication* 6, 2 (1969), 3–13. https://doi.org/10.1177/002194366900600202

[20] U. Hahn and I. Mani. 2000. The challenges of automatic summarization. *Computer* 33, 11 (2000), 29–36. https://doi.org/10.1109/2.881692

[21] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. arXiv:1801.06146 [cs.CL]

[22] Bin Ke and Santhosh Ramalingegowda. 2005. Do institutional investors exploit the post-earnings announcement drift? *Journal of Accounting and Economics* 39, 1 (2005), 25–53. https://doi.org/10.1016/j.jacceco.2004.02.002

[23] Dongcheol Kim and Myungsun Kim. 2003. A Multifactor Explanation of Post-Earnings Announcement Drift. *The Journal of Financial and Quantitative Analysis* 38, 2 (2003), 383–398. http://www.jstor.org/stable/4126756

[24] J. Peter Kincaid et al. 1975. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel.

[25] George R Klare. 1974. Assessing readability. *Reading research quarterly* (1974), 62–102.

[26] S.P Kothari. 2001. Capital markets research in accounting. *Journal of Accounting and Economics* 31, 1 (2001), 105–231.

[27] G. Harry Mc Laughlin. 1969. SMOG Grading-a New Readability Formula. *Journal of Reading* 12, 8 (1969), 639–646. http://www.jstor.org/stable/40011226

[28] Tim Loughran and Bill Mcdonald. 2011. When Is a Liability NOT a Liability? Textual Analysis, Dictionaries, and 10-Ks. *The Journal of Finance* 66 (02 2011), 35 – 65. https://doi.org/10.1111/j.1540-6261.2010.01625.x

[29] John D. Lyon, Brad M. Barber, and Chih-Ling Tsai. 1999. Improved Methods for Tests of Long-Run Abnormal Stock Returns. *The Journal of Finance* 54, 1 (1999), 165–201. http://www.jstor.org/stable/222413

[30] Walaa Medhat et al. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* 5, 4 (2014), 1093–1113.

[31] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *CoRR* abs/1310.4546 (2013). arXiv:1310.4546 http://arxiv.org/abs/1310.4546

[32] Jeffrey Pennington et al. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[33] Matthew E. Peters et al. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Association for Computational Linguistics, Florence, Italy, 7–14. https://doi.org/10.18653/v1/W19-4302

[34] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, Vol. 31.

[35] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. https://doi.org/10.48550/ARXIV.1908.10084

[36] Guilherme Moraes Rosa et al. 2021. To Tune or Not to Tune? Zero-Shot Models for Legal Case Entailment. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law* (São Paulo, Brazil) *(ICAIL '21)*. New York, NY, USA, 295–300. https://doi.org/10.1145/3462757.3466103

[37] Matthias Schnaubelt and Oleg Seifert. 2020. Valuation Ratios, Surprises, Uncertainty or Sentiment: How Does Financial Machine Learning Predict Returns From Earnings Announcements? *SSRN Electronic Journal* (01 2020). https://doi.org/10.2139/ssrn.3577132

[38] RJ Senter and Edgar A Smith. 1967. *Automated readability index*. Technical Report. Cincinnati Univ OH.

[39] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. https://doi.org/10.48550/ARXIV.2004.09297

[40] Kai Ming Ting and Ian H Witten. 1999. Issues in stacked generalization. *Journal of artificial intelligence research* 10 (1999), 271–289.

[41] Ashish Vaswani et al. 2017. Attention Is All You Need. https://doi.org/10.48550/ARXIV.1706.03762

[42] David H. Wolpert. 1992. Stacked generalization. *Neural Networks* 5, 2 (1992), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

[43] Zhengxin Joseph Ye and Björn W. Schuller. 2021. Capturing dynamics of post-earnings-announcement drift using a genetic algorithm-optimized XGBoost. *Expert Systems with Applications* 177 (2021), 114892.

[44] Zhengxin Joseph Ye and Björn W. Schuller. 2021. Deep Learning Post-Earnings-Announcement Drift. In *2021 International Joint Conference on Neural Networks (IJCNN)*. 1–7. https://doi.org/10.1109/IJCNN52387.2021.9534436