Forecasting S&P 500 Returns with Earnings Calls Transcripts

Authors: Alessandra Di Giacomo, Francesco Bellotto, Francesco Carlo Genito

***Abstract*—This paper examines the use of earnings call transcripts to predict S&P 500 returns. By analyzing the textual content of these calls, which offer insights into company performance, the study looks into the effectiveness of transformer-based models like DistilBERT, BERT, and Longformer for forecasting. These models generate embeddings from the transcripts, which are then aggregated using mean pooling or Principal Component Analysis (PCA) to predict the S&P 500 return for the following month. The findings show that Longformer with PCA aggregation yields the best predictive accuracy. Overall, the study demonstrates how earnings call data can be valuable for forecasting broader market trends.**

I. Introduction

This project focuses on forecasting the S&P 500 returns, using insights extracted from the transcripts of the earnings calls held by the companies. Earnings calls offer the company management the opportunity to share financial results and projections, offering a valuable source of information for analysts and investors. Using these transcripts, this work aims to find out whether the content disclosed in these conferences conveys any information on the general macroeconomic status. The analysis combines data from different financial datasets, including WRDS (Wharton Research Data Services) [1] and FRED-MD [2], which provides information on financial, accounting, and economic data. We developed our analysis using a structured approach, exploiting data preprocessing, feature extraction, and exploratory data analysis. Then, it proceeds towards the comparison of how different transformer-based models like `DistilBERT` [3], `BERT` [4], and `Longformer` [5] are able to predict the values of the S&P 500 return, using the earnings calls transcripts. This analysis not only sheds light on the informational value of earnings calls but also contributes to understanding their potential influence on macroeconomic trends.

II. Data Analysis

The dataset has been created by merging data taken from tables of WRDS. Firstly, we selected companies in the S&P 500 index, taking *permno* and *gvkey* as identifiers. We used the former to obtain from CRSP (Center for Research in Security Prices) the Market Capitalization, a key indicator of a company's size and the latter to extract the relevant earnings calls' transcripts from Compustat IQ. The plot in Figure 1 illustrates the general trend of Market Capitalization for the ten S&P 500 companies with the lowest economic size during the period from January 1, 2009, to December 31, 2020. As shown in Figure 1, the composition of S&P 500 can vary each quarter, as the index's membership is updated to reflect

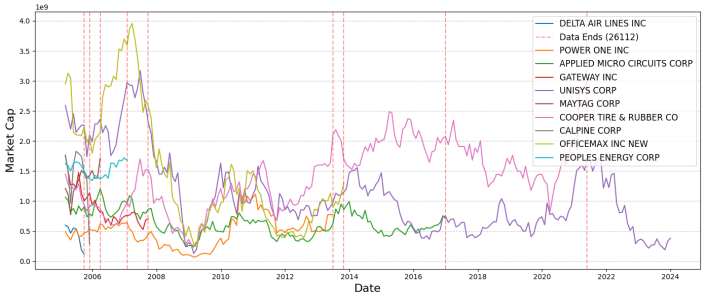

Fig. 1. Market Capitalization evolution of the bottom 10 companies in the S&P 500 Index. (Red line if the company left the index).

the current ranking based on market capitalization. Hence, we decided to opt for the choice of all the 500 companies with the largest average Market Value, without excluding the ones that left the index during the selected period (see VII for details). After this examination, we have proceeded with an exploratory data analysis for the transcript texts, by starting from the visualization of the box-plot in Figure 2 and showing which texts are way too short/long, (i.e. texts which fall outside the range $[Q1 - 1.5 * IQR; Q3 + 1.5 * IQR]$, where $Q1$ is the first quartile, $Q3$ is the third one and $IQR = Q3 - Q1$ is the interquartile range), and we decided not to consider them in the predictions. The output data, taken from FRED-

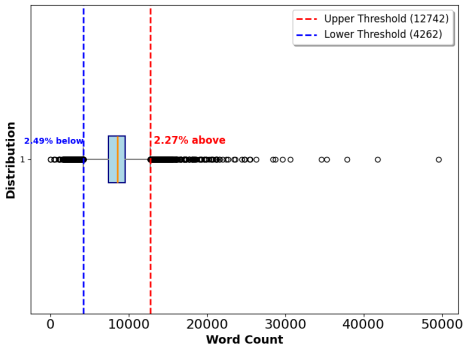

Fig. 2. Boxplot of Word Counts

MD, will be instead associated to the S&P 500 return value, a key measure of market performance, which represents the monthly percentage change in the S&P 500 price. This return is computed using the following formula:

$$S\&P500_{ret,t} = \frac{S\&P500_{price,t} - S\&P500_{price,t-1}}{S\&P500_{price,t-1}} \quad (1)$$

III. Methods and Models

A. Methods

We employed transformer-based models to analyze earnings call transcripts and predict the S&P 500 return. We used `DistilBERT`, `BERT`, and `Longformer`, which are three language models based on the Transformer architecture with the goal of analyzing textual objects and the context of each word inside it. Their strategy consists of splitting the texts into small fragments, known as *tokens*, and computing the self-attention matrix, which combines different information between tokens in order to get a numerical representation of the context. Finally, tokens and attention masks are given as input to a transformer-specific pre-trained model, in order to get the final embeddings of each transcript text. `BERT` model (Bidirectional Encoder Representations from Transformers) is characterized by the analysis of a single word context both from left and right, thanks to the bidirectional self-attention. Furthermore, in order to improve the training performance, `BERT` is pre-trained using Masked Language Modeling (MLM), a process which consists of masking some words in a sentence, and letting the model predict them, and Next Sentence Prediction (NSP), which predicts whether two sentences follow each other [4]. `DistilBERT` is a faster and slimmer version of `BERT`, as it contains half the amount of transformers' layers, it inherits only the MLM training task from `BERT` and it applies a knowledge distillation to replicate its performance [3]. `Longformer` also uses MLM but replaces global attention with Local Attention, allowing it to process much longer sequences (up to 4096 tokens), instead of the 512 tokens limit given by the `DistilBERT` and `BERT` architectures. The Table I summarizes the key differences between the three transformers.

To focus on the most representative features, we retained, for each of these three models, only the embedding of the [CLS] (classification) token for each sequence, a special token added at the beginning as a summary representation of the entire input text, making it ideal for tasks like classification or, in our case, regression.

Aspect	BERT (base)	DistilBERT	LongFormer
Layers	12	6	12
Hidden Size	768	768	768
Parameters	110M	66M	150M
Train Tasks	MLM + NSP	MLM	MLM + Local Attention
Max Tokens	512	512	4096

TABLE I
TRANSFORMERS SUMMARY COMPARISON

Since multiple earnings calls occur during a single quarter, we needed to aggregate embeddings for all the conferences within the same period. We explored two approaches:

- MEAN POOLING : we computed the mean of the [CLS] token embeddings for all texts within a given quarter, producing a single embedding that represents the information available in the entire period.
- PRINCIPAL COMPONENT ANALYSIS (PCA) [6] : after computing the covariance matrix for the embeddings in each quarter of the training set, we extracted its first principal components (the chosen number was considered an hyperparameter) and aggregated the data projection onto these on a unique vector. For the validation and test sets, the adopted principal components were the ones corresponding to the last training quarter's ones.

The PCA identifies and assigns greater weight to the directions that explain the most variance. This approach effectively captures the most meaningful information while reducing dimensionality, resulting in a compact and efficient representation that retains the critical features of the original embeddings.
The final embedding for each quarter served as input a for a neural network (NN) with a regression output layer. The neural network is a fully connected feedforward architecture with a funnel-shaped structure designed to progressively reduce the feature space. It begins with an input layer corresponding to the input variable size, followed by several hidden layers where the number of neurons decreases at each step, with ReLU activation functions applied between layers. Specifically, the first hidden layer maps the input features to a representation of size *hidden_size*. The subsequent hidden layers gradually reduce the number of neurons: the first layer to $\frac{hidden\_size}{2}$, the second to $\frac{hidden\_size}{4}$, and the third to $\frac{hidden\_size}{8}$. The final output layer maps the last hidden layer (with $\frac{hidden\_size}{8}$ neurons) to a single continuous output value. We selected the optimal *hidden_size* using a validation approach and we updated it for each training period, accordingly to the new input data. Weight decay has been applied to each model as regularization technique.

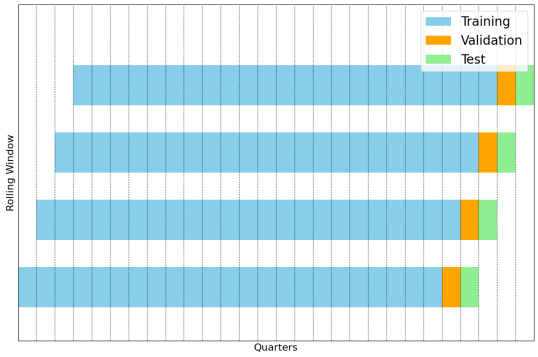

Fig. 3. Train, Validation and Test Sets for the Time Series

B. Models

Our main task is to use all the transcripts associated to the earnings calls of a certain quarter to predict the S&P 500 return value associated to the immediately next month (e.g we will predict the output value of Apr 2009 relying on the textual embeddings between Jan 2009 and Mar 2009).

Then, since we have data available from the period spanning from Jan 2009 to Dec 2020, we identified the right trade-off between the length of the training period and the number of available testing periods. For example, training on the whole period of available texts will result in having only one final test value; on the other hand, reducing each training period to only one quarter and moving forward will result in having very few information to learn from, possibly leading to underfitting. Hence, to handle financial data and time series while preserving the chronological order of information, we employed a rolling window validation approach. We construct the dataset and train and validate and test with it, as shown in Figure 3. Specifically, each rolling window consists of the following structure:

- The training set covers the first 23 quarters (5 years and 9 months).
- The validation set corresponds to the subsequent quarter.
- The test set includes the quarter immediately following the validation set.

Model 1 - Baseline: The baseline leverages `DistilBERT` as a transformer to generate text embeddings. For each quarter, the embeddings are aggregated by calculating their component-wise mean. This is followed by a single-layer neural network with a linear activation function, effectively reducing the model to a linear regression for prediction. Given the simplicity of the architecture (a single layer with linear activation), the learning rate is the primary hyperparameter tuned during validation.

Model 2 - DistilBERT with Mean Pooling: Starting from the baseline approach, we enhanced the model's complexity by incorporating non-linearity into the neural network and adding 3 hidden layers. The model continues to leverage `DistilBERT`, with mean pooling applied across each quarter. To limit overfitting, we applied weight decay with a value of 0.01 as a regularization technique. The learning rate and the starting hidden size were tuned following our rolling basis validation approach. By doing this, the model's hyperparameters were optimized for each time window. Additionally, we set a maximum number of 30 epochs and stop the training process if the train loss remains stable for more than 3 consecutive epochs, with a tolerance of 10^{-4} for the improvement.

HS/LR	10^{-4}	$3 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	10^{-3}	$3 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
128	0.0224	0.0009	0.00035	0.00006	0.00007	0.00006
256	0.00053	**0.00002**	0.00038	0.00075	0.00024	0.00013
512	0.00099	0.00018	0.00022	0.00021	0.00056	0.00028
1024	0.00102	0.00075	0.00032	0.00027	0.00036	0.00015

TABLE II
VALIDATION LOSSES ON FIRST WINDOW

Table II presents the training and validation losses for all possible combinations of hyperparameters, specifically for the initial training window (Jan 2009 - Sep 2014). It is important to note that the optimal hyperparameters may vary across different training phases due to the rolling window. Therefore, the table reflects only the performance metrics corresponding to the first training process.

Model 3 - BERT with Mean Pooling: We applied all the steps described in III-B to the embeddings created with `BERT`. To make a consistent comparison with the first model, we used the same architecture for the neural network and the same validation procedure. Figure 4 shows the predicted values against the real S&P 500 returns.

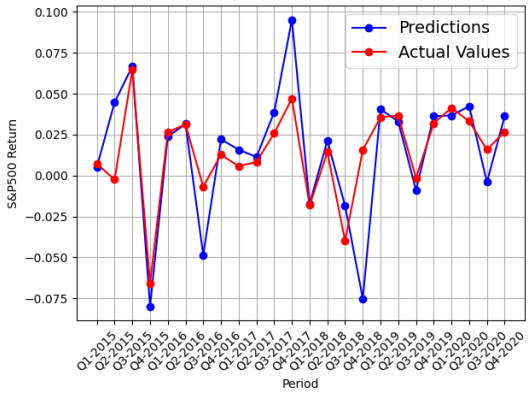

Fig. 4. Predicted output against actual output using `BERT` with mean pooling.

Model 4 - Longformer with Mean Pooling: We follow all the steps described in III-B to the embeddings created using `Longformer`, using the same NN and the same hyperparameters' values for the validation.

Model 5 Longformer - with PCA : The model in III-B outperforms the results obtained with `DistilBERT` and the ones obtained with `BERT`. Therefore, we retained `Longformer` as transformer and repeated the whole procedure described above. Instead of a mean aggregation, we adopted a Principal Component approach to combine the information for each quarter.

Specifically, we applied PCA to the input embeddings using different numbers of principal components and then concatenated them into a single final vector. To manage the larger input size, we used larger hidden sizes during validation to avoid excessively shrinking the input dimensions from the first layer. Table III shows the metrics obtained for the different numbers of principal components selected.

	R^2	MSE	RMSE	Mean Explained Var.
1 Components	-0.753	0.00133	0.0365	44.26%
5 Components	0.243	0.00058	0.0240	91.40%
10 Components	**0.413**	**0.00045**	**0.0211**	**95.89%**

TABLE III
PERFORMANCE COMPARISON FOR DIFFERENT NUMBERS OF COMPONENTS.

As shown in Table III, PCA produces a more informative aggregated representation for each quarter by preserving the most significant patterns and relationships within the embedded texts. In this framework, the number of principal components used plays a crucial role. Using only one component explains only 44.26% of the total variance and, as a result, the final

performance is worse than simply using the mean of the output observations, as indicated by the negative R^2. On the other hand, 5 components explain on average more than 90% on the total variance and outperforms the mean aggregation. The performance further improves by including 10 components, meaning that having more granular information leads to more accurate results. Figure 5 shows the predictions against the actual output values.

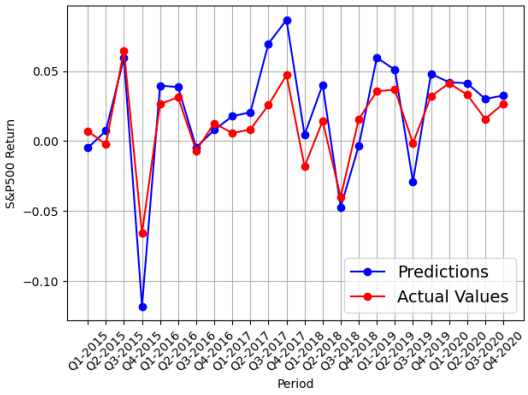

Fig. 5. Predicted output against actual output using `Longformer` with PCA.

IV. Results

Table IV shows a summary of the main metrics used to evaluate the performance of the different models. The R^2 score evaluates the proportion of the variation in the output variable that is explained by the transcripts in input. The mean squared error measures the average squared difference between predicted values and actual values, indicating the model's prediction accuracy. These coefficients are defined as:

$$R^2 = 1 - \frac{\sum_{n=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{n=1}^{N}(y_i - \bar{y})^2} \qquad MSE = \frac{1}{N}\sum_{n=1}^{N}(y_i - \hat{y}_i)^2$$

where y_i are the actual values, $\bar{y}$ is their average, and $\hat{y}_i$ are the predicted value.

	R^2	MSE	RMSE
Baseline	-0.572	0.0012	0.0346
DistilBERT - Mean	-0.385	0.0011	0.0325
BERT - Mean	0.105	0.00068	0.0261
Longformer - Mean	0.266	0.00056	0.0236
Longformer - PCA	**0.413**	**0.00045**	**0.0211**

TABLE IV
Performance comparison for different models.

Table IV compares the performance of the different models for predicting S&P 500 and it validates the visual results in Figure 4 and Figure 5. The baseline model shows the worst performance, with a negative R^2 which indicates that it performs worse than simply predicting the mean return. A first enhancement is achieved by using a more complex NN's architecture and by introducing non-linearity in the model. However, DistilBERT's reduced model size and simpler architecture limit its ability to effectively capture the intricate patterns necessary for accurate predictions. This results in relatively poor performance, as indicated by its negative R^2 More powerful models like BERT or Longformer, which can handle larger contexts and deeper relationships in the text, are more suitable for this task and provide better predictive accuracy.

The most significant improvement is seen when PCA is used to aggregate the embeddings. Longformer with PCA aggregation achieves both the lowest MSE and the highest R^2, indicating the best predictive accuracy and explanation power. This suggests that PCA not only helps to improve model performance by reducing the error but also ensures a more robust and informative representation of the data than doing a simple component-wise mean of the available embeddings.

V. Limitations

The main issues encountered with our project were related to the structural limits of the transformers' architecture. Firstly, our transformers-based models automatically truncate the transcripts texts to take into account only the first 512 tokens if using `DistilBERT` or `BERT` (4096 for `LongFormer`). Hence, a portion of each text had been neglected, keeping only information contained in the first section of each transcript. However, during earnings calls, the most important information is typically contained inside the first portion of the text, where main topics of the speech are highlighted. As a consequence, the maximum length limitation does not infer drastically on the performance. Another important limitation stems from the decision of keeping only the classification token [CLS] for each transcript to make our final predictions. This may lead to use fewer details from each earnings call, since the classification token summarizes the whole information of the total amount of embedded tokens. Nevertheless, the approach of using only this informative token is widely used in contexts of semantic regression [7], in order to combine a good capture of meaningful information with a lower computational cost.

VI. Conclusion

This study explored the use of earnings call transcripts to predict S&P 500 return values using advanced transformer-based models like DistilBERT, BERT, and Longformer, along with aggregation techniques such as mean pooling and PCA. The Longformer model paired with PCA achieved the best performance, demonstrating that these transcripts hold valuable information to forecasting market trends.

Despite limitations such as text truncation and reliance on [CLS] token embeddings, critical insights in the initial section of the transcripts, and a rolling window validation approach helped maintain prediction reliability. The findings highlight the potential of unstructured text data in financial forecasting. This work showcases the value of text-based predictive models in financial machine learning, providing a foundation for extracting meaningful insights from unstructured data.

VII. Ethical Risk Assessment

While our approach has the potential to improve the accuracy of forecasts and provide valuable information for investors, it also presents various ethical risks that must be considered.

An important risk is the fairness of the solution. Indeed, one of the potential risks is the exploitation of earnings calls to provide biased information aimed at positively influencing their own market positions. As a result, the data used to train the models may be biased, favoring certain companies, sectors, regions, which could result in inaccurate predictions that disproportionately benefit specific groups or markets. Once being aware of the problem, we decided to proceed with our exploratory data analysis to understand better the distribution of the smallest companies inside the S&P 500 index. Then, we trained our model on data obtained by considering all the S&P 500 companies, averaged along the train period, so as not to privilege largest companies over the smallest ones, and so, considering into our analysis also the companies which left the S&P 500 index during the selected period.

Lastly, the limited access to data also leads to some privacy concerns. Earnings call transcripts may contain sensitive company information that could be misused if not handled properly. Strict data anonymization and encryption practices should be implemented to ensure that private data remains secure and in compliance with regulatory standards.

VIII. Appendix

We add here the plots of the predicted values against the actual S&P 500 returns for the other models, respectively the baseline 6, `DistilBert` with mean pooling 7 and `Longformer` with mean pooling 8.

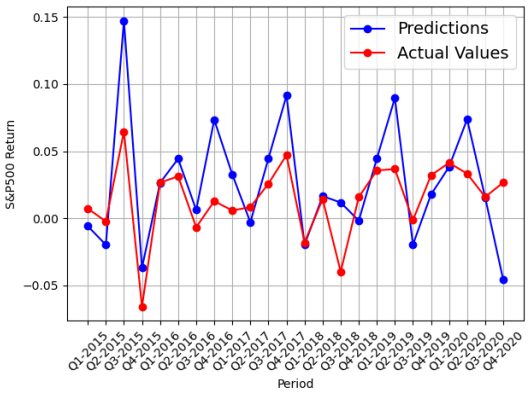

Fig. 6. Predicted output against actual output using baseline model

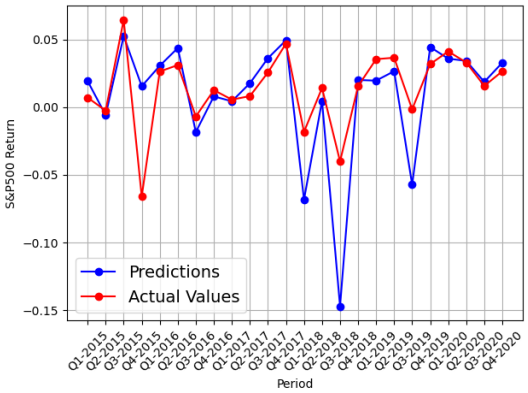

Fig. 7. Predicted output against actual output using `DistilBERT` with mean pooling.

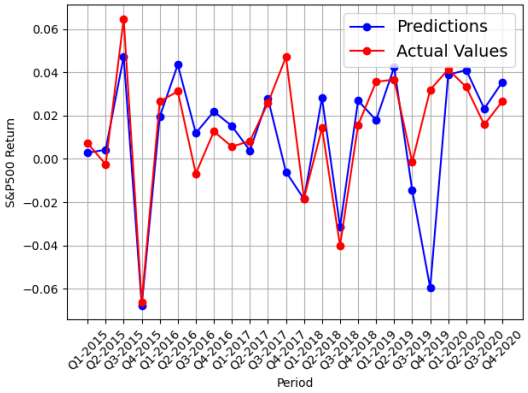

Fig. 8. Predicted output against actual output using `Longformer` with mean pooling.

References

[1] "Wharton research data services (wrds)," https://wrds-www.wharton.upenn.edu, 2024.

[2] M. W. McCracken and S. Ng, "Fred-md: A monthly database for macroeconomic research," Federal Reserve Bank of St. Louis, Working Paper 2015-012B, August 2015, revised version. [Online]. Available: https://doi.org/10.20955/wp.2015.012

[3] H. Face, "Distilbert model documentation," 2024. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/distilbert#transformers.DistilBertModel

[4] ——, "Bert model documentation," 2024. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/bert

[5] ——, "Longformer model documentation," 2024. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/longformer

[6] S. Raschka, "Principal component analysis in 3 simple steps," 2015, accessed: 2024-12-19. [Online]. Available: https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html

[7] H. Wagner, "Generalized semantic regression using contextual embeddings," 2023. [Online]. Available: https://www.thebigdatablog.com/generalized-semantic-regression-using-contextual-embeddings/