Online Appendix of:

The News in Earnings Announcement Disclosures: Capturing Word Context Using LLM Methods

Federico Siano
University of Texas at Dallas
federico.siano@utdallas.edu
Phone: 972-883-4994

Naveen Jindal School of Management
800 W Campbell Rd, Richardson, TX 75080

February 2025

Online Appendix

Methodological Details: Textual parsing of tables and generic statements

I split each textual disclosure into sentences using Python and the NLTK library (https://www.nltk.org/). I test the sentence tokenization algorithm for 30 disclosures and find a 93% parsing accuracy. I then classify a sentence as "table" whenever it contains at least 20 non-breaking spaces (i.e., "\xa0"), dash symbols (i.e., "-"), or plus symbols (i.e., "+"). I exclude sentences classified as "tables" from the main analyses (see also Section 3 of the paper).

I identify and exclude generic cautionary statements (see Section 3 of the paper for rationales and robustness checks) using regular expressions that match multiple tokens (e.g., "Cautionary Statements", "Forward-looking Statements"). See below one example of cautionary statements:

Ocean Bio-Chem Inc., August 14th, 2017 (CIK: 0000350737)

"*Certain statements contained in this Press Release including without limitation, Company performance in the second half of 2017, the Company's entry into the pet market and commencement of production in the expanded portion of the Company's plant, constitute forward-looking statements. For this purpose, any statements contained in this report that are not statements of historical fact may be deemed forward-looking statements. Without limiting the generality of the foregoing, words such as "believe," "may," "will," "expect," "anticipate," "intend," "could" including the negative or other variations thereof or comparable terminology are intended to identify forward-looking statements.*"

Since long documents include relatively more tables and cautionary or generic statements, both with non-standard formatting (thus more likely to go undetected using the prior algorithms), I focus on the first 1,500 tokens for disclosures with an above-average token count to enhance prediction accuracy. I also conduct the main analyses using two alternative approaches: (a) *without* excluding any tokens, and (b) excluding only the last 512 tokens for disclosures with an above-average token count. I find a moderately lower power for *CAR LLM EA*, but the tenor of the results does not change. Note that 512 tokens represent the maximum sequence length that the *BERT* LLM can process. In the following paragraph, I outline the method used to overcome this limitation and process the entire earnings announcement text.

Methodological Details: BERT LLM implementation

Among recent *BERT* LLMs, I choose the *RoBERTa* implementation (i.e., Robustly Optimized BERT Pretraining Approach; Liu et al., 2019) because of its high accuracy and wide availability. Specifically, I fine-tune a pre-trained (*RoBERTa*) *BERT*-based LLM that can be freely downloaded from *GitHub* (https://github.com/pytorch/fairseq/tree/main/examples/roberta). I use the "RoBERTa Large" model and the PyTorch framework, implemented via the "Simple Transformers" Python library, for fine-tuning.[1]

I fine-tune the *BERT*-based LLM using a "learning rate" of 1e-6 and 3 "training epochs".[2] I use a "hold-out" sample of 500 earnings press releases (i.e., a validation set never used neither for training nor for out-of-sample prediction) to select the best hyperparameters using a grid search algorithm. The average computational time for fine-tuning is *3 hours* per training epoch (i.e., about *9 hours* in total) using an NVIDIA Tesla A100 GPU accessible through *Google Cloud Platform*. The out-of-sample modeling/prediction time amounts to *1 hour* for each outcome of interest.

BERT LLMs can process textual sequences of maximum 512 tokens. To overcome this limit and characterize disclosures more completely, I divide earnings press releases longer than 512 tokens into "windows" or subsequences of (maximum) 512 tokens. Each "window" or subsequence overlaps with the prior "window" or subsequence for 20% of the tokens (i.e., I define a "stride" of 80%).[3] Each generated subsequence is used for fine-tuning purposes. During out-of-sample modeling, the *BERT*-based LLM outputs a prediction for each "window" or subsequence. I compute the arithmetic mean of the relevant predictions.

---

[1] I opt for a *RoBERTa* "Large" (as opposed to a "Base") model to maximize the modeling of contextual connections across parts of speech, that benefit from a higher number of trainable parameters. Notably, current and widely accessible computational resources make *RoBERTa* "Large" fine-tuning feasible.

[2] The *learning rate* represents the size of the step through which a loss or cost function is minimized. Too high learning rates cause the global minimum of the cost function to be missed in the optimization process, while too large ones increase computational time and render the model's training unfeasible. The *training epochs* are the number of times that the entire training dataset is passed through the neural network's artificial neurons. Epochs are needed to minimize the models' loss function through an iterative process of gradient descent. The choice of the number of epochs is critical: too few iterations do not allow the model to properly minimize the loss function; too many iterations lead to in-sample overfitting and low out-of-sample accuracy.

[3] The "stride" is the distance chosen to slide the window when generating textual subsequences. Smaller strides allow the model to learn from a larger number of textual sequences but require higher fine-tuning time.

Methodological Details: Alternative gradient boosting modeling of text and reported numbers

I follow Frankel et al. (2022) and model disclosures using one and two-word phrases (i.e., "N-Grams"). I pre-process text in 5 steps; I (i) convert text into lowercase characters, (ii) remove "stop-words" included in the "NLTK" library, (iii) remove numbers and punctuation, (iv) lemmatize words,[4] and (v) select the top 1,500 one and two-word "N-Grams" based on the TF-IDF protocol.[5] I finally use "N-Grams" as inputs to a non-linear Gradient Boosting model to predict, through disclosure text, *CAR [0,1]* around earnings announcement dates.

I choose a "Gradient Boosting" modeling approach because it outperforms alternative techniques, including "Random Forest", "Support Vector Regressions", and "Supervised Latent Dirichlet Allocation" (see Section 3). I implement Gradient Boosting using 2,500 regression trees, and a 0.01 learning rate.[6] In all instances, I use a training/prediction protocol and corresponding observations that are identical to those used for *BERT*-based fine-tuning and out-of-sample prediction. I employ a widely available *Python* software implementation: "LightGBM Gradient Boosting"[7] For alternative modeling approaches (e.g., "Random Forest", see Section 3) I choose the hyperparameters following Frankel et al. (2016 and 2021). Notably, the results remain robust when using 3,000 or 5,000 regression trees; however, performance generally deteriorates as the number of trees increases.

I apply a Gradient Boosting protocol (with the same hyperparameters described above) to predict, through the financial statement surprises, *CAR [0,1]* around earnings announcement dates.

---

[4] "Lemmatization" is a common pre-processing technique in natural language processing that allows to link semantically and morphologically similar words to a common underlying construct. For example, the words "report" and "reports" would be lemmatized as "report". In sensitivity analyses, I also use "stemming", a technique that removes words suffixes but ignores the morphological role of words: I find similar results.

[5] TF-IDF (i.e., term frequency-inverse document frequency) is a natural language processing method to evaluate how relevant is a word in a document based on its occurrence in a larger collection of documents. More infrequent words are considered more relevant and are thus given more weight. Research shows that this approach produces a better regression fit in natural language processing tasks (Loughran and McDonald, 2016). Since Frankel et al. (2022) use all "unigrams" and "bigrams", I also test the sensitivity of my results to selecting top 3,000 and 9,000 "N-Grams" by TF-IDF: I find similar results.

[6] I use a "hold-out" sample of 500 earnings press releases (i.e., a validation set never used neither for training nor for out-of-sample prediction) to select the best hyperparameters using a grid search algorithm.

[7] Available at https://lightgbm.readthedocs.io/en/stable/.

Examples of Influential Words

Tercile 3 of CAR [0,1]

Microsoft Corp., Jul. 19th, 2016 (CIK: 789019): *CAR [0,1]*: +3.5%; *CAR LLM EA*: +2.5%

[1] *"During the quarter, Microsoft returned $6.4 billion to shareholders in the form of share repurchases and dividends."*

[2] *"Office consumer products and cloud services revenue grew 19% (up 18% in constant currency) with Office 365 consumer subscribers increasing to 23.1 million."*

[3] *"The current quarter effective tax rate reflected a favorable mix of our income between the U.S. and foreign countries, as well as benefits associated with distributions from foreign affiliates."*

Tercile 2 of CAR [0,1]

BlackRock Inc., Jan. 16th, 2019 (CIK: 1364742): *CAR [0,1]*: +1.9%; *CAR LLM EA*: +0.6%

[1] *"4% increase in full year revenue driven by growth in base fees and technology services revenue, partially offset by lower performance fees"*

[2] *"Restructuring charge of $60 million from initiative to modify the size and shape of the workforce excluded from as adjusted results."*

[3] *"BlackRock generated total net inflows of $124 billion in 2018. This included $50 billion of fourth quarter net inflows and record quarters for iShares and illiquid alternative strategies."*

Tercile 1 of CAR [0,1]

Lululemon Athletica Inc., Mar. 26th, 2020 (CIK: 1397187): *CAR [0,1]*: -4.7%; *CAR LLM EA*: -1.6%

[1] "*In March 2020, we temporarily closed all of our retail locations in North America , Europe , Malaysia , New Zealand , and we temporarily closed our distribution center in Sumner, WA. These locations currently remain closed."*

[2] *"Due to the impact that COVID-19 is having across the globe, and the rapid and continuous developments, we are not providing guidance for fiscal 2020 at this time."*

[3] *"We are now navigating an extraordinary environment, which is currently impacting our business."*

The above examples apply the SHAP algorithm ("*SHapley Additive ExPlanations*"; Lundberg and Lee, 2017) to identify the most influential words contributing to the LLM's predictions. Three examples are provided, one for each tercile of *CAR [0,1]*. Words enclosed in squares are the most influential, with green color indicating a positive contribution to the prediction and red color indicating a negative contribution.

Table OA-1: Sensitivity Analyses

Panel A: Sample Outside of the LLM's Pre-Training Period (Ending 2019)

	Dependent Variable: *CAR [0,1]*					
	Fine-Tuning: 2006-2013 | Analysis: 2021-2023			**Fine-Tuning: 2021-2022 | Analysis: 2023**		
	(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
CAR LLM EA	**0.97** ***		**0.75** ***	**1.12** ***		**0.87** ***
CAR GB EA		0.40 ***	0.01		0.47 ***	0.04
CAR GB FSA		0.71 ***	0.49 ***		0.80 ***	0.58 ***
N	27,853	27,853	27,853	9,503	9,503	9,503
Adjusted R^2	14.3%	11.8%	17.8%	15.8%	13.7%	20.6%
Incr. Adj R^2			**6.0%*** **			**6.9%*** **

Panel B: Monthly Fama MacBeth Regressions

	Dependent Variable: *CAR [0,1]*			
	Cross-sectional Estimations		**Time-series Average**	
Percentage of month-years	**(1)**	**(2)**	**(3)**	**(4)**
w/ ***R^2 of CAR LLM EA* > 0.10**	**82%**			
w/ ***Incr R^2 of CAR LLM EA* > 0.05** (or $\Delta R^2 > 50\%$)		**75%**		
CAR LLM EA				**0.79** ***
CAR GB FSA			0.88 ***	0.59 ***
Average Adjusted R^2			13.3%	21.4%

Panel C: Impact of Additional Financial Statement Variables

	Dependent Variable: *CAR [0,1]*			
	Linear Modeling		**ML Gradient Boosting Modeling**	
	Adj R^2	**Incr R^2**	**Adj R^2**	**Incr R^2**
	(1)	**(2)**	**(3)**	**(4)**
CAR LLM EA	**14.9%**		**14.9%**	
Textual Attributes + Financial Surprises + Other FSA Items	6.0%	**10.4%** ***	12.9%	**6.0%** ***

This table presents sensitivity and robustness tests. Panel A analyzes a sample outside the LLM pre-training period (ending July 2019), comparing the explanatory power of *CAR LLM EA* against Gradient-Boosting measures (*CAR GB EA* and *CAR GB FSA*). Panel B presents Fama-MacBeth regressions results from 288 monthly panels spanning 2000–2023. The *BERT*-based LLM is fine-tuned annually using a 5-year backward-rolling window and a random 25% of yearly observations. Columns (1) and (2) highlight the number of month-years where *CAR LLM EA* achieves an adjusted and incremental R^2 exceeding 10% and 5%, respectively, reflecting a 50% average improvement in explanatory power. Columns (3) and (4) report the average of monthly adjusted R^2. Panel C compares the explanatory power of *CAR LLM EA* to regressions enhanced with additional financial statement analysis (FSA) items. These FSA items include dividends and their quarterly changes, leverage and its quarterly changes, restructuring charges, and analyst following. Continuous variables (excluding LLM and other machine-learning-based predictions) are winsorized at the 1st and 99th percentiles of their quarterly distribution, and regressions are estimated without fixed effects. Statistical significance at <0.01, <0.05, and <0.10 is denoted by ***, **, and *. Incremental adjusted R^2 significance is evaluated using 10,000 bootstrap iterations, with all variables detailed in the Appendix.

Table OA-2: Predicting the Level and Uncertainty of Future Quarterly Earnings

Panel A: Text in Press Releases and Future Earnings (*N* = 97,355)

	Dependent Variable: *Earn* $_{Q+1}$				
	(1)	**(2)**	**(3)**	**(4)**	**(5)**
Adj R^2					
Earn LLM	**39.6%**				
Textual Attributes		8.4%			
Financial Statement Surprises			34.9%		
Earn GB EA				16.6%	
Earn GB FSA					39.2%
Incr R^2	**-**	**31.1% *****	**11.0% *****	**23.2% *****	**6.3% *****

Panel B: Text in Press Releases and Future Earnings Volatility (*N* = 91,648)

	Dependent Variable: *SD Earn* $_{Q+1-Q+4}$				
	(1)	**(2)**	**(3)**	**(4)**	**(5)**
Adj R^2					
SD Earn LLM	**30.4%**				
Textual Attributes		4.9%			
Financial Statement Surprises			25.2%		
SD Earn GB EA				12.8%	
SD Earn GB FSA					28.6%
Incr R^2	**-**	**24.6% *****	**12.1% *****	**19.0% *****	**8.2% *****

This table presents the absolute and incremental adjusted-R^2 ("Adj R^2" and "Incr R^2", respectively) of *BERT*-based LLM predictions of the level and uncertainty of future quarterly earnings. Panel (A) [B] presents results for the prediction of (next-quarter earnings, *Earn* $_{Q+1}$; *N* = 97,355 firm-quarter observations) [next-four-quarters standard deviation of earnings, *SD Earn* $_{Q+1-Q+4}$; *N* = 91,648 firm-quarter observations]. In Panel (A) [B], the experimental variable is (*Earn LLM*) [*SD Earn LLM*], which represents the predicted dependent variable, (next-quarter earnings) [next-four-quarters standard deviation of earnings], from a LLM that uses only the text of a firm's quarterly earnings press release. *(SD) Earn GB EA* [*(SD) Earn GB FSA*] represents predicted next-quarter earnings (next-four-quarters standard deviation of earnings) obtained from a Gradient Boosting model that is trained based on one and two-word phrases, also known as "unigrams" and "bigrams" [that is trained based on financial statement surprises].

Continuous variables (excluding LLM and other machine-learning-based predictions) are winsorized at the 1st and 99th percentiles of their quarterly distribution. ***, **, * indicate significance at <0.01, <0.05, <0.10. The statistical significance of the incremental adjusted-R^2 is assessed using a bootstrapping protocol based on 10,000 iterations (with replacement). All variables are described in the Appendix.

Table OA-3: *BERT* LLMs' Accuracy, Textual Attributes, and Firms' Fundamentals

	Dependent Variable: *Accuracy*		
	(1)	**(2)**	**(3)**
Tone	-0.03 *		-0.08 ***
	(0.07)		(0.00)
Fog	0.03 ***		0.03 ***
	(0.00)		(0.00)
Length	0.12 *		0.11 **
	(0.07)		(0.01)
Numbers	0.01 ***		0.01 ***
	(0.00)		(0.00)
Future	-0.13 **		-0.16 ***
	(0.03)		(0.01)
Earn		4.40 ***	4.04 ***
		(0.00)	(0.00)
Earn Surp		-0.00	-0.00
		(0.19)	(0.19)
Sales Surp		-1.60 *	-1.16
		(0.06)	(0.16)
Div		1.90 ***	1.77 ***
		(0.00)	(0.00)
Leverage		-0.54 ***	-0.51 ***
		(0.00)	(0.00)
Restr		-0.40 ***	-0.40 ***
		(0.00)	(0.00)
Spec Items		-0.08 ***	-0.08 ***
		(0.00)	(0.00)
Size		0.10 ***	0.11 ***
		(0.00)	(0.00)
N	98,171	98,171	98,171
Fixed Effects	No	No	No
Adjusted R^2	0.01	0.04	0.05

This table examines to what extent and how *BERT* LLMs' accuracy, for the task of predicting contemporaneous *CAR [0,1]*, varies with narrative attributes and firm fundamentals. The dependent variable is *Accuracy* which is calculated as the absolute difference between actual and *BERT* LLM predicted *CAR [0,1]*, multiplied by -1. Continuous variables are winsorized at the 1st and 99th percentiles. Standard errors are clustered by firm and quarter-year. *p*-values are reported in parentheses. ***, **, * indicate significance at <0.01, <0.05, <0.10. All models are estimated with an intercept (unreported). All variables are described in the Appendix, with the exception of *Earn* (quarterly earnings scaled by lagged market value of equity), *Div* (quarterly dividend per share), *Leverage* (quarterly leverage ratio), and *Restr* (quarterly restructuring expenses).