Context, Language Modeling, and Multimodal Data in Finance

Sanjiv Das, Connor Goggins, John He, George Karypis, Sandeep Krishnamurthy, Mitali Mahajan, Nagpurnanand Prabhala, Dylan Slack, Rob van Dusen, Shenghua Yue, Sheng Zha, and Shuai Zheng

Sanjiv Das
is a professor of finance at Santa Clara University and an Amazon scholar at Amazon Web Services in Santa Clara, CA and Palo Alto, CA.
srdas@scu.edu

Connor Goggins
is a software development engineer at Amazon Web Services in Palo Alto, CA.
cggoggin@amazon.com

John He
is a software development engineer at Amazon Web Services in Palo Alto, CA.
hezhijia@amazon.com

George Karypis
is a professor of computer science at the University of Minnesota and senior principal scientist at Amazon Web Services in Palo Alto, CA.
gkarypis@amazon.com

Sandeep Krishnamurthy
is a software development manager at Amazon Web Services in Palo Alto, CA.
krsandee@amazon.com

Mitali Mahajan
is a graduate student at Santa Clara University in Santa Clara, CA.
mmahajan@scu.edu

Nagpurnanand Prabhala
is a professor of finance at Johns Hopkins University in Baltimore, MD.
prabhala@jhu.edu

Dylan Slack
is a graduate student at University of California in Irvine, CA.
dslack@uci.edu

Rob van Dusen
is a graduate student at the University of Chicago in Chicago, IL.
rvanduse@chicagobooth.edu

Shenghua Yue
is a software development engineer at Amazon Web Services in Palo Alto, CA.
yuesheng@amazon.com

Sheng Zha
is a senior applied scientist at Amazon Web Services in New York, NY.
zhasheng@amazon.com

Shuai Zheng
is an applied scientist at Amazon Web Services in Palo Alto, CA.
shzheng@amazon.com

KEY FINDINGS

- Machine learning based on multimodal data provides meaningful improvement over models based on numerical data alone.
- Context-rich models perform better than context-free models.
- Pretrained language models that mix common text and financial text do better than those pretrained on financial text alone.

ABSTRACT

The authors enhance pretrained language models with Securities and Exchange Commission filings data to create better language representations for features used in a predictive model. Specifically, they train RoBERTa class models with additional financial regulatory text, which they denote as a class of RoBERTa-Fin models. Using different datasets, the authors assess whether there is material improvement over models that use only text-based numerical features (e.g., sentiment, readability, polarity), which is the traditional approach adopted in academia and practice. The RoBERTa-Fin models also outperform generic bidirectional encoder representations from transformers (BERT) class models that are not trained with financial text. The improvement in classification accuracy is material, suggesting that full text and context are important in classifying financial documents and that the benefits from the use of mixed data, (i.e., enhancing numerical tabular data with text) are feasible and fruitful in machine learning models in finance.

TOPICS

Quantitative methods, big data/machine learning, legal/regulatory/public policy, information providers/credit ratings*

In recent papers, Araci (2019), Desola, Hanna, and Nonis (2019), and Yang, Uy, and Huang (2020) showed that improvements in language models using additional training with financial text resulted in better prediction of sentiment of news headlines (i.e., financial phrases). In this article, we assess whether language models improve on the simple use of text-based numerical features (e.g., sentiment, readability, positivity, negativity, riskiness, litigiousness; see Loughran and McDonald 2020). The existing practice in much of the finance literature is to create word-based features by applying

*All articles are now categorized by topics and subtopics. **View at PM-Research.com**.

finance-specific dictionaries such as the Loughran–McDonald (LM) word lists.[1] Such word scoring leads to numerical features that are widely used in regression analysis of regulatory filings, tweets, news, and so on. These features have proven to be quite successful, and a vast literature has exploited them. This article shows that the recent developments in language modeling, such as bidirectional encoder representations from transformers (BERT), can extend these successes.

Text is more versatile, extensive, and multifaceted than mere tabular (numerical) data (Gentzkow, Kelly, and Taddy 2019). Furthermore, numerical time-series data may be less useful given the huge structural shifts in economies from recent major events such as trade wars and pandemics. Therefore, finance companies are exploring the advantages of using text, which is plentiful in finance and forward-looking. Examples of use cases are as follows:

- Credit analysis of firms using text in their Securities and Exchange Commission (SEC) filings and news. Additional text improves models based purely on tabular data. This is useful to financial institutions with a lending business, rating agencies, and managers of corporate bond funds who would like to use text to enhance their credit quality models for asset allocation (see Bodnaruk, Loughran, and McDonald 2015; Bonsall et al. 2017; and Ertugrul et al. 2017).
- Asset managers using text features for prediction and portfolio construction. Using text features enables companies to discover other similar companies (e.g., semantically similar firms). In the opposite case, semantic distance may be used for diversification of portfolios. Machine learning (ML) using big data was used by Routledge (2019).
- Classification models based on text are being used to predict corporate performance. Various textual features, such as sentiment, readability, tone, size, risk, and uncertainty, have been established as useful in rank ordering the future performance of firms, and these are table-stakes features that companies employ in text processing to make portfolio decisions.[2] Comparison of text across quarters and years offers useful predictive information, as shown by Cohen, Malloy, and Nguyen (2020).

These use cases are examples that have been shown to provide better performance than pure tabular data in several academic studies and industry white papers.[3] Several papers also provide comprehensive surveys of textual analysis in finance, and an excellent recent survey is given by Loughran and McDonald (2020).[4]

Against this backdrop of the text mining literature in finance and accounting, we next discuss the research questions in this article.

RESEARCH QUESTIONS

An advantage of word-based scoring to create numerical variables is that the dimension of the feature set remains small and supports simple, explainable methods such as regression analysis. This is why it has been popular and practical among the financial community, both in academia and in practice. The addition of text poses

[1] See sraf.nd.edu/textual-analysis/resources/.

[2] See Antweiler and Frank (2004), Das and Chen (2007), Hoberg and Phillips (2016), and Bach et al. (2019).

[3] For more examples, see Hafez et al. (2020a, 2020b), Cong, Liang, and Zhang (2019), Cong et al. (2019), and Chebonenko, Gu, and Muravyev (2018).

[4] Other surveys are provided by Li (2010b), Das (2014), Kearney and Liu (2014), Loughran and McDonald (2016), and Gentzkow, Kelly, and Taddy (2019).

interesting technical problems, the two most salient being (1) how to combine text and numerical tabular data in one predictive model; and (2) language models such as BERT being easiest to apply to short text (i.e., a few sentences) and, in fact, working best at the single sentence level.

Adapting classifiers to handle long text is an important problem and has only recently been getting attention. Long documents are ubiquitous in finance, and regulatory filings are just one example—there are financial news reports, analyst reports, and so on, all of which run into hundreds of words. Therefore, using language models such as BERT as they are is unlikely to be effective given that input is restricted to maximum sequence lengths that mostly vary from 128 to 1,024 tokens and more often are on the lower side. This is good for news headlines and tweets but not good for reports and long documents, which have been the subject of suggested improved models (e.g., Liu et al. 2018 and Lee and Hsiang 2019 for patents, Wan et al. 2019 for legal documents, and Pappagari et al. 2019 for transcripts of earnings calls, all of which are widely used in natural language processing [NLP] applications).

Most recently, generating long sequences with sparse attention transformers also has been gaining traction (Child et al. 2019; Zaheer et al. 2020). Sparse transformers typically deal with documents with sequence lengths of several thousands. SEC filings, however, generally have more than 50,000 words, and this can be doubled after byte-pair encoding (BPE) tokenization. Some of the experiments here will deal with long documents, and we will try to address these issues using the simplest approaches that give good results. Given this, we consider the following research experiments in this article:

1. Using a benchmark dataset of financial news headlines (composed of one or two sentences), from the work of Malo et al. (2014) and Araci (2019), a comparison of word-based scoring and full language modeling determines how important context is in the classification of new sentiment. This is redone on another dataset as an additional experiment, which examines whether the full text in SEC 10-K/Q and 8K filings contains predictive information beyond that from word-based features. These experiments examine whether adding text to tabular data generally improves classification in the finance domain.
2. We conduct a comparison of general-purpose language models of the BERT class (BERT, RoBERTa, DistilBERT) versus a language model specifically trained on financial regulatory filings (which we denote RoBERTa-Fin) to assess whether pretrained financial language models add value to general purpose financial models. This has already been explored in work by Araci (2019), who undertook further pretraining of BERT using a subset of the Reuters TRC2 news articles dataset. Here, the article experiments with a model that is pretrained only on SEC 10-K/Q filings and another that is trained on SEC filings and Wiki text, both of which contain formal financial text. In this experiment, a RoBERTa model with dynamic masking and no next sentence prediction (NSP; used in BERT) is used in the pretraining scheme.
3. We assess how these models perform when predicting filing week returns, a classification exercise that is hard in efficient markets. Markets are by and large efficient, and time-series approaches to predicting stock movements have been rather unsuccessful, even at high frequency (Malkiel 2003). However, a growing literature has found that quarterly earnings and stock performance rankings are weakly predictable.[5]

[5] See Antweiler and Frank (2004); Bodnaruk, Loughran, and McDonald (2015); Bonsall et al. (2017); Brown and Tucker (2011); Bushee, Gow, and Taylor (2018); Das and Chen (2007); Ertugrul et al. (2017); Hoberg and Phillips (2016); Jegadeesh and Wu (2013); Li (2010a); Li and Zhao (2014); Loughran, McDonald, and Yun (2009); Loughran and McDonald (2011, 2013, 2014, 2015); Price et al. (2012); and Tetlock, Saar-Tsechansky, and Macskassy (2008).

4. We conduct experiments to assess whether long documents can be classified accurately using document summaries instead. This is especially important because the signal in numerical variables may be swamped by noisy text in long documents. Using a summary may reduce the noise, but it also loses context, which can degrade classification. This interesting trade-off is explored briefly.

DATA

Standard Features

The SEC 10-K and 10-Q filings for all tickers in the S&P 500 index every quarter for the past 10 years (2010–2019) were downloaded. These data come from SEC's open data platform, EDGAR, which has been shown to improve the informativeness of markets; see Gao and Huang (2020). This delivered over 20,000 documents. These documents were used to pretrain a RoBERTa-Fin language model.

As is standard practice in finance, one may wish to focus on a limited portion of the SEC 10-K/Q text (i.e., the management discussion and analysis (MD&A) section) because this section is where management provides a forward-looking analysis of the company's financials. Therefore, this section was parsed out as well.

We also downloaded the financial phrase bank data prepared by Malo et al. (2014) to undertake experiments on financial news headlines to assess the value of context. This dataset contains three sentiment classes—negative, neutral, and positive—and thus offers a good test bed for comparing word-based NLP classification and language model–based classification in the finance domain.

Finally, to create another dataset, we downloaded some 10-K/Q and 8K forms for 645 tickers of companies that took Paycheck Protection Program (PPP) loans; see Balyuk, Prabhala, and Puri (2020). This resulted in 5,810 filings for 2020 Q1 and Q2, for which we examine filing week return performance.

Pretraining a Language Model

What are language models? Language models are collections of conditional and unconditional probabilities of words in sequences of words. Therefore, a language model is a probability distribution over sequences of words. More accurately, given a sequence of words, we can compute the probability of all those words occurring in that sequence. The key word here is *sequence*—this provides context that we assume to be important.

We use the 10-K/Q filings data for the past decade of S&P 500 tickers to pretrain a RoBERTa class model (Liu et al. 2019). For pretraining, the entire text of the 10-K/Q filing was used, not just the MD&A section, to ensure a broader context of financial language was captured. For classification, passing the original documents through a language model generates new word embeddings that reflect context-transformed data, which may then be passed into another neural network for the final classification task.

Pretrained BERT (Devlin et al. 2019) models and variants such as RoBERTa (Liu et al. 2019) have been proven to achieve state-of-the-art results on a wide variety of NLP tasks, including question answering (SQuAD v1.1), natural language inference, and others. The pretrained BERT models generate bidirectional representations for sentences; that is, they assign probabilities to forward and backward sequences of words. The representations can be used as additional features or can be finetuned in downstream NLP tasks.

The BERT model is pretrained with the book corpus and English Wikipedia. Araci (2019; FinBERT), Beltagy, Lo, and Cohan (2019; SciBERT), Lee et al. (2019; BioBERT), and Huang, Altosaar, and Ranganath (2019; ClinicalBERT) showed that using a BERT model pretrained with a domain-specific corpus may improve the accuracy of domain-specific NLP tasks. For this reason, we have trained two models, RoBERTa-Fin1 and RoBERTa-Fin2, the former using only the last decade's S&P 500 10-K/10-Q reports and the latter using both, as well as the last decade's filings and Wikipedia text. The original RoBERTa model is pretrained on Wikipedia, book corpus, openweb text, and stories, but for RoBERTa-Fin2 we only used Wiki text because we do not want the finance text to be overwhelmed by other data sources. In particular, we used dynamic masking and removed NSP loss in the pretraining scheme of RoBERTa. The methodology we describe here may be used to pretrain other language models in the finance domain using proprietary text as well.

To train these models with more than 100 million parameters, a number of graphics processing units (GPUs) were required. We used eight Amazon Web Services EC2 p3dn.24xlarge instances. Each instance has 92 vCPUs and eight Nvidia V100 Tensor core GPUs. As a result, we needed a distributed deep learning training framework to run the pretraining job using all GPUs on the eight instances. Pretraining data were distributed into 256 shards. Using dynamic masking, the training instances with random masked tokens were generated on the fly during pretraining. We selected Uber's Horovod framework[6] with openMPI[7] on top of Nvidia Collective Communications Library.[8] A representation learning script from the Amazon Web Services GluonNLP[9] library was used for pretraining. See Exhibit 1 for a schematic of the pretraining model. We trained two versions, base and large; the former takes between two and seven days of pretraining, and the total number of training iterations is 250,000 with a minibatch size of 2,048 and sequence length of 512.[10]

Summarization

One approach to handling long documents is to summarize the document first and then submit it to a classifier. This has potential, especially if the summary keeps signal and drops noise in the document. However, it may also lose context. Furthermore, the type of the summary may matter, and we will discuss these issues in light of the following approach to summarization, called a *Jaccard summarizer*. This summarizer is based on breaking the original large document into separate sentences. Once this is done, the Jaccard similarity[11] is computed for each pair of sentences. If there are n sentences, then the number of similarity computations is $n(n-1)/2$. We sum the Jaccard values to get n total similarity scores. The summary is then the top sentences from a ranked list based on similarity scores. This summary distills the document to sentences that are most similar to all pf the other sentences. We may further reweight the similarity scores based on special lexicons (word lists) and then

[6] https://github.com/horovod/horovod.

[7] https://www.open-mpi.org/.

[8] https://developer.nvidia.com/nccl.

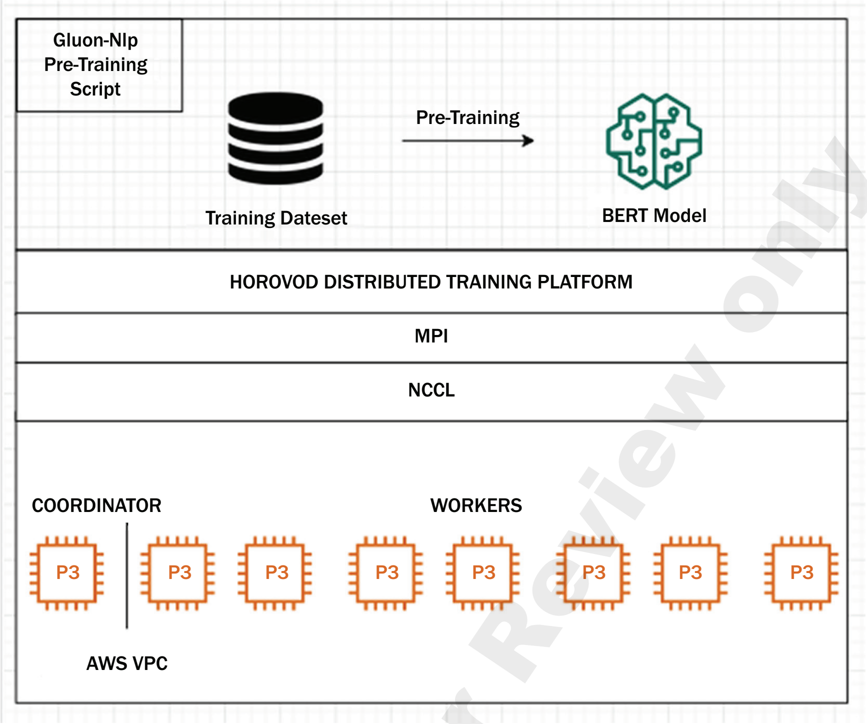
[9] https://gluon-nlp.mxnet.io/examples/word_embedding/index.html.

[10] The very first step in applying BPE encoding is to split the word or input into characters. We then use the merge operations learned during BPE training to combine pairs recursively. If there is no match in the merge operations, the input will remain in its split form, as a sequence of characters. For example, word embeddings may be split into subword tokens [em, ##bed, ##ding, ##s]. This is also one reason why is it hard to model the SEC filings; the number of tokenized tokens may be more than 100,000 (i.e., very large).

[11] The size of the intersection set of word tokens from each sentence divided by the size of the union set.

EXHIBIT 1

Schema for Pretraining RoBERTa-Fin Models



NOTES: The data type used was float16, batch size was 2,048, learning rate was 0.000625, number of training steps was 250,000, optimizer was LANS (Zheng et al. 2020), warm-up ratio was 125%, maximum sequence length was 512, and number of Horovod slots per instance was 8. We used a vocabulary size of 32,000. We trained two RoBERTa-Fin models, with embedding sizes of 768 and 1,024.

select top sentences based on weighted similarity scores. For example, we may weight sentences using financial word lists, specifically the litigious words. Or we may use a lexicon of commonly used words in equity markets to get a summary that is geared toward the equity analyst, or fixed-income lexicons for the bond analyst.

RESULTS AND FINDINGS

In this section, we present the various experiments that were undertaken and the results and implications therefrom.

How Much Does Language Matter?

This section evaluates the sentiment classification of financial news sentences using different approaches for comparison. There is a vast literature using the LM word lists to score text to convert it into numerical variables for regression analysis. Loughran and McDonald (2020) specifically presented a review of several papers that use text features such as readability, sentiment, polarity, positivity, negativity, risk/uncertainty, and litigiousness, which are converted into numerical scores. These effectively eliminate language from the model specification. To assess how much language matters, we use the financial phrase bank dataset from Malo et al.

(2014). quantify news headlines into numerical scores, and build an ML model to classify sentiment using only the numerical scores. We then enhance the model with text to see how much language matters. The dataset is especially appropriate for this experiment because the text in each headline is short and thus has minimal context, even though it exceeds that of the numerical scores.

We used 2,264 documents. The dataset has three categories of hand-tagged sentiment labels. These are imbalanced: 303 negative, 1,391 neutral, and 570 positive. We scored each news article for readability (Gunning-Fog index). Using the LM word lists, we also scored the percentage of litigious, risk/uncertainty, negative, and positive words. Finally, we scored polarity, which is the difference in positive and negative words divided by the sum of both. As is typical in ML, we created normalized features from these variables using min-max scaling. This created six features in the dataset, and we dropped the column of text containing the news headlines. Using word vectors from FastText,[12] we also created an alternative scoring mechanism with a different construction of word lists, comprising the following word roots: positive, negative, certainty, uncertainty, litigious, risk, safe, and fraud. These are also used separate from the LM word lists and result in slightly higher accuracy on the same classification task because of the additional features beyond those of LM.

To avoid making and justifying choices in the ML model, we decided to implement an open-source AutoML tool out of the box. This has the benefit of eliminating modeler bias, in which the choice of model may bias the outcome of the experiment. Another useful aspect of AutoML modeling is that the model usually searches for the best ensemble of individual ML models that give good results; thus, we try to find the best model given the numerical features first, so as to set a difficult baseline before we add full text to the analysis. We chose to use AutoGluon Tabular,[13] an open-source package. This is easy to implement and replicate. It runs an ensemble of models, such as random forest, Gini-based and entropy-based extra trees classifiers, K-nearest neighbors, CatBoost, NN, and light GBM. It is also fast and has high performance (see Erickson et al. 2020). AG-Tabular took less than 20 seconds to fit the best model it could find. The accuracy on the test dataset with the LM word scores is 66% with a Matthews (1975) correlation coefficient (MCC) of 0.18 (this lies in the range of –1 to +1, where 0 means no classification ability and –1 means perfect error always). Detailed results are reported in Exhibit 2. In addition, when we used our new features based on fastText, we had a small improvement in accuracy to 71%, and each other metric showed similar improvement. We note that, although AutoGluon allows additional options, such as hyperparameter tuning and autostacking of models, we did not select these to prevent these decisions being interpreted as modeler bias.

With purely numerical features, the model achieved a 2/3 prediction accuracy on the test dataset.

We now include an additional column with the text containing the original financial sentences from the news. Once again, AG-Tabular was applied and took a runtime of under 30 seconds. AG-Tabular converts text columns into TFIDF embeddings that are combined with the numerical columns and then submitted to the classifier. It applied the same models to the data and achieved an 86% accuracy level on the test dataset with an MCC of 0.72, compared with an accuracy of 66% and MCC of 0.18 when only numerical features were used. The other statistics are reported in Exhibit 2.

It is clear that the metrics are good across all classes. Although we get over 2/3 accuracy with just numerical text attributes, we can see that using all of the text (full

[12] fasttext.cc/from Facebook.

[13] autogluon.mxnet.io/.

EXHIBIT 2

Comparison of Classification Performance on Tabular Data and TabText for the Sentiment Classification Task on the Financial Phrase Bank

	Tabular Only		
Label/Metric	**(LM)**	**(New)**	**TabText**
Accuracy	0.658	0.711	0.861
MCC	0.186	0.447	0.723
Negative			
Precision	0.625	0.625	0.973
Recall	0.089	0.149	0.643
F1	0.156	0.241	0.774
Neutral			
Precision	0.670	0.754	0.854
Recall	0.966	0.918	0.989
F1	0.791	0.828	0.917
Positive			
Precision	0.481	0.598	0.838
Recall	0.121	0.563	0.626
F1	0.194	0.580	0.717

NOTES: We report accuracy, precision, recall, F1 scores, and MCC on the test (holdout) dataset. When we add text to the dataset (denoted as TabText), we see an appreciable improvement in the classification. The LM column uses numerical scores based on LM word lists, and the column New uses word lists based on our word vector model.

language) matters in classifying financial news in this dataset, and we do better than when only using the numerical text scores.

Classification Using SEC Forms 10-K, 10-Q, and 8-K

The experiments in the previous section have shown that using all of the text improves on using numerical NLP scores from text (e.g., readability, risk) in a classification setting. We move on to classification experiments on large amounts of text: SEC filings. We extracted all SEC 10-K, 10-Q, and 8-K forms for the first two quarters of 2020 for a list of 645 tickers. These tickers were based on those used by Balyuk, Prabhala, and Puri (2020) and are related to firms that borrowed money provided by the federal government in the CARES Act of 2020 under the provisions of the PPP. This is not entirely necessary for our analysis, and we may just as well have chosen any random set of tickers; however, this is an interesting set because these firms may have something in common given that they all partook in the PPP program. Our SEC form extractor took 45 minutes to download and create a dataframe of 5,801 forms, of which 530 are 10-Ks, 704 are 10-Qs, and 4,567 are 8-Ks.

We also downloaded the stock prices for all tickers from December 2019 through July 2020, so as to span 2020 Q1 and Q2 with a month on each side. After converting these to returns, we lined up the returns for each filing from five days before the filing date to five days after the filing date (11 days). We did the same for the S&P 500 return, and we also computed the excess return of the ticker over the S&P 500 return for all 11 days. We created a new variable: the sum of daily excess returns from day 0 through day 5 (i.e., from the day of the filing and the five following days, roughly a week of post-filing performance). Using this new variable, we created a binary label, taking a value of 1 if the variable was positive and 0 otherwise, for each row of the dataset; the ratio of variables is 0.45:0.55, respectively—fairly well balanced.

First, we submitted the combination of the long text of the SEC filings and the day 5 through day 1 returns for the ticker and the S&P 500 to AutoGluon for classification of the binary label. We implemented an 80/20 train/test split for the dataset. We achieved an accuracy of 58.5% in predicting five-day cumulative returns using the filing, with an MCC of 0.14, which is not high, although it is reasonable given that predicting stock movements is difficult in efficient markets.

Second, using a subset of 440 tickers, some of these firms first took the PPP loans and then returned the money. The reasons for returning the money and the cost–benefit trade-off of keeping and returning the money were analyzed in detail by Balyuk, Prabhala, and Puri (2020). Reasons for returning the money may include avoiding regulatory oversight, preventing markets from penalizing stock prices because taking a PPP loan sends a negative signal, or simply realizing ex post that the firm did not need the funding and returning the money would send a positive signal to the markets with a concomitant boost to the stock price. Our initial extraction of all filings for the 440 tickers in 2020 Q1 and Q2 resulted in 4110 filings.

EXHIBIT 3

Predicting Filings by Companies That Returned PPP Money

Label/Metric	Tabular Only	TabText
Accuracy	0.83	0.89
MCC	0.25	0.57
Label 0		
Precision	0.83	0.88
Recall	0.99	0.99
F1	0.91	0.93
Label 1		
Precision	0.79	0.93
Recall	0.10	0.40
F1	0.19	0.56

NOTES: There were 4,100 original filings. Dropping post-return filings yielded 3,191 with label 0 and 720 with label 1, totaling 3,911. This means that a model that always returns label 0 will have an accuracy of 0.816. The tabular data only comprise cumulative returns up to the filing data, and the TabText data contain tabular data and the full text of the filing.

For the subsample of 440 tickers of firms that returned the money, we applied a label of 1 for filings by firms that returned the money and 0 for those firms that did not. We also dropped all filings of companies that returned the money, starting from the date of the filing that stated the money was being returned, so as to not keep any information on or after the date on which the money was returned.

Using this dataset, we then conducted experiments to predict label 1 (companies that returned the PPP money) or 0 (those that did not) using the following two feature sets: (1) only cumulative returns on days −5 to 0 or (2) cumulative returns on days −5 to 0, plus the full text of the filing. We also computed the cumulative daily return from day 0 to day +5. We found that for firms that did not return the money (label 0), the mean day −5 to day 0 return was 3.5%, but it was only 0.2% for filings of firms that returned the money. This relationship reverses post-filing: Day 0 to day +5 return was 0.01% for firms that did not return the money but 0.28% for firms that returned the money. This suggests that, as a first cut, stock returns before filing dates may distinguish between firms that returned the money and those that did not, because the two groups appear to have different average returns before and after the filing dates in the cross section of firms. Given that past returns are helpful, we further explore whether the text of the filing improves the ability to classify the text of firms that returned PPP money and those that did not.

The prediction performance is reported in Exhibit 3. The baseline accuracy is 81.6%, the return-only model has an accuracy of 83%, and the returns plus full text model achieves 89% accuracy, suggesting that adding text to tabular data improves classifier performance. The TFIDF approach handled long text well for these filings and improved classification accuracy for mixed numerical tabular and text models.

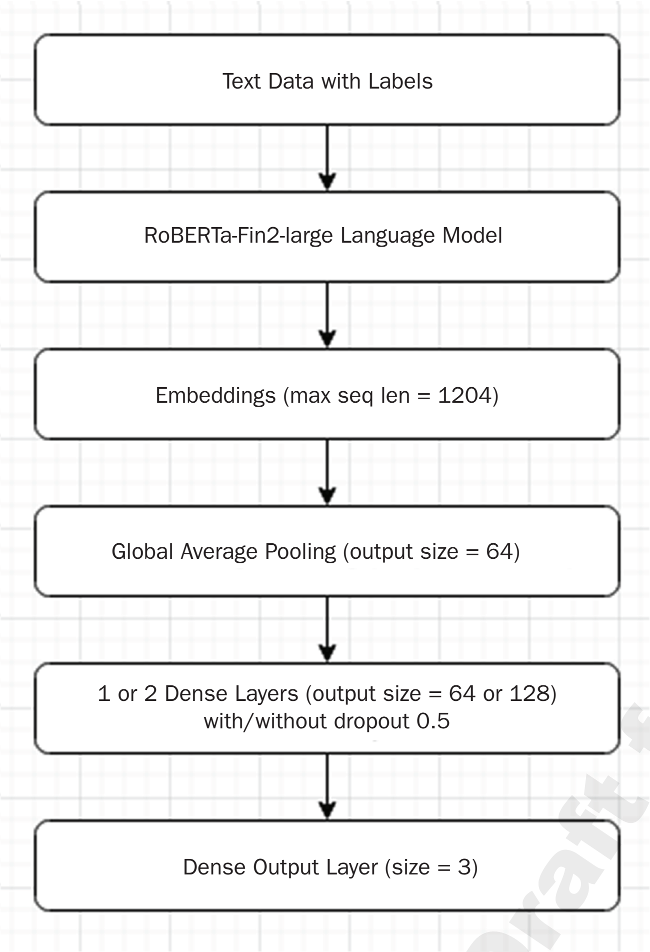
Adding Context with Language Models?

We now explore whether language models that have context can do even better than models that do not. A classifier with *n*-gram features that are not cognizant of context offers a baseline. The fact that we achieved 2/3 accuracy with just six numerical features (Exhibit 2), however, suggests that the large literature on textual scoring using keywords and lexicons is well placed. However, we saw that exploiting all of the text using TFIDF makes a huge difference in classification accuracy out of sample (increasing it from 66% to 86%), so we know that adding text *n*-grams does not result in massive overfitting and improves model performance over mere NLP scoring. The question is, can we do even better with the addition of context from transformer-based (Vaswani et al. 2017; Devlin et al. 2019; Liu et al. 2019; Sanh et al. 2020), language models?

We assess different language models here, from the original BERT model to variants such as DistilBERT and RoBERTa. We also assess pretrained models, RoBERTa-Fin1, pretrained completely on the SEC 10-K/Q filings for S&P 500 companies for all quarters in the 2010–2019 period. We also pretrained a language model on Wikipedia text and all the text used in RoBERTa-Fin1; we denote this model as RoBERTa-Fin2. We will assess both base- and large-model variants.

EXHIBIT 4

Schematic of the Models Used after Applying BERT Models to the Text and Passing the Embeddings through an Average Pooling Layer


Text Data with Labels
RoBERTa-Fin2-large Language Model
Embeddings (max seq len = 1204)
Global Average Pooling (output size = 64)
1 or 2 Dense Layers (output size = 64 or 128) with/without dropout 0.5
Dense Output Layer (size = 3)

NOTE: We then use one or more dense layers of size 64 or 128 with and without dropout (0.5).

The vast array of language models is used to exploit *transfer learning*. That is, we use the pretrained language model to modify the text input and create embeddings that are then passed into one or two additional layers in a neural network for training the specific classification task—in this case, scoring sentiment into three categories. We consider five different architectures for comparison across all the language models and feed the embeddings for text from the language model into standard feed-forward neural net layers. Rather than feeding the embeddings for only the CLS token, as is done with standard BERT models, we applied average pooling to the embeddings for all tokens and submitted that to the following architectures:

- Configuration 1: one dense layer (64), baseline model
- Configuration 2: one dense layer (128)
- Configuration 3: two dense layers (64, 64)
- Configuration 4: one dense layer (64), one dropout layer (0.5)
- Configuration 5: one dense layer (128), one dropout layer (0.5)

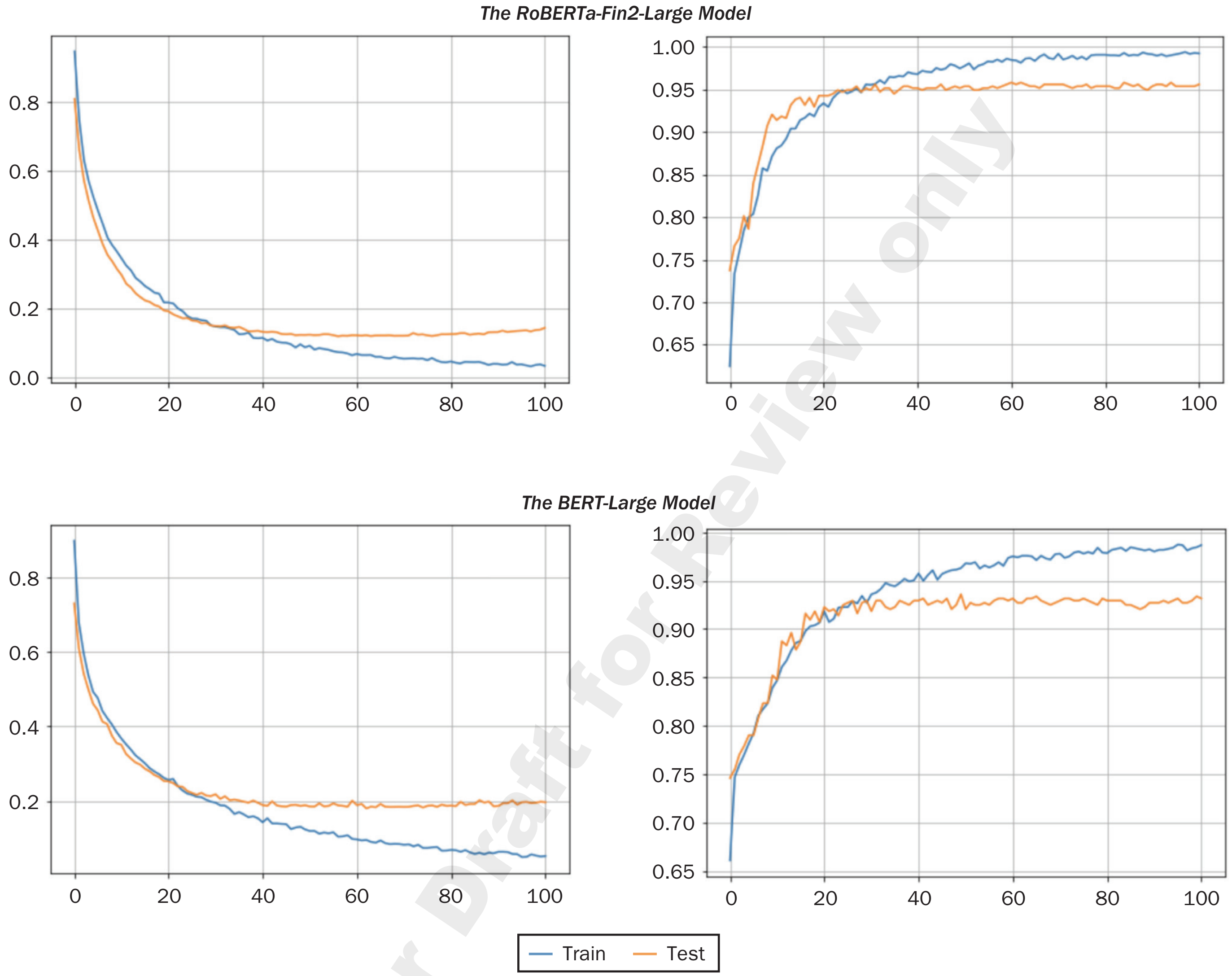
See Exhibit 4. All configurations have a final dense output layer of three neurons for the final classification. The dropout layers help in managing overfitting. We trained each model for 100 epochs and report the accuracy level on the test dataset for 30, 50, and 100 epochs. Exhibit 5 shows how the loss function and accuracy track with epoch for the training and validation datasets for the top two models. The RoBERTa-Fin2-large model has slightly higher accuracy and overfits marginally less than BERT-large.

Results for the financial news sentiment classification are presented in Exhibit 6. The language models clearly outperform TFIDF, which is an ensemble model that uses context-free textual features. This is the case even given that NLP scores for readability, sentiment, and so on were used in the classifier and not used in the language model–based classifiers. Therefore, for financial text, context certainly matters, consistent with similar results in the legal (Lee and Hsiang 2019; Wan et al. 2019), biological (Lee et al. 2019), and science (Beltagy, Lo, and Cohan 2019) domains.

The pretrained RoBERTa class models, tuned for financial text, are of two types: (1) pretrained using only SEC filings data for the past decade (RoBERTa-Fin1) and (ii) pretrained using both the SEC data and Wiki text (RoBERTa-Fin2). A comparison of these models enables us to assess whether making the language model have only finance context and no other will improve its performance in classifying financial text. As we can see from a comparison of our new language models' performance (RoBERTa-Fin1 versus RoBERTa-Fin2) including both financial and nonfinancial context does improve the model over one that only has financial context—RoBERTa-Fin2 (both base and large) outperforms RoBERTa-Fin1. In fact, in comparison with RoBERTa-Fin1, traditional BERT models do better, which suggests that financial context alone may

EXHIBIT 5

Progress of Training and Validation Loss and Accuracy for the RoBERTa-Fin2-Large Model and the BERT Large Model for Comparison


The RoBERTa-Fin2-Large Model

The BERT-Large Model

NOTES: The RoBERTa-Fin2-large model pertains to the top two plots, and the BERT-large model is shown in the bottom two plots. The left-side plots display the loss function, and the right-side plots display model accuracy. We see that the accuracy levels out at around 30 epochs. At 100 epochs, it is 95.6% for the RoBERTa-Fin2-large model and 93.2% for standard BERT-large.

be too narrow for good classification and the best route is to further pretrain general language models with additional financial text. Finally, we also see that RoBERTa-Fin2 outperforms language models such as BERT that do not have extended pretraining with financial text, but the gains here are smaller. In sum, context matters most, and financial context matters incrementally.

Finally, we examine use of summarized text. It is likely that we may be able to improve classifier performance by using a summary of the text instead of all the text. This has the potential to remove some of the noise in the text and to avoid truncation of the text when it gets too long. We redid the experiment by replacing the full text of the SEC filings with the summary of the text. We did not find that summarization helps improve metrics (precision, recall, F1). Therefore, we conclude that the use of text summaries for classification gives ambiguous results, and more testing is needed to assess the efficacy of summaries.

EXHIBIT 6

Comparison of Performance on the Sentiment Classification Task across Various Language Models

	Accuracy/Configuration No.		
Model	**30 Epochs**	**50 Epochs**	**100 Epochs**
AutoGluon		86.1%	
Distilbert hf	92.9%/4	92.9%/4	93.2%/4
Distilbert gluon	92.3%/1	92.3%/2	92.9%/4
BERT Base	92.1%/2,5	92.3%/4	93.4%/4
BERT Large	93.2%/4	92.3%/3,4	92.3%/3
RoBERTa Base	79.2%/1,2	81.2%/5	80.6%/5
RoBERTa Large	85.7%/3	86.1%/3	86.1%/2,5
RoBERTaFin1 Base	90.3%/4	90.3%/1,5	90.5%/2
RoBERTaFin1 Large	92.9%/2	92.9%/5	92.7%/1
RoBERTaFin2 Base	93.6%/2	94.3%/3	94.7%/2,3,5
RoBERTaFin2 Large	95.6%/3	95.6%/2,5	95.6%/1

NOTE: We report the results using various models on the same dataset and for the five neural net configurations mentioned in the text.

CONCLUDING DISCUSSION

In this article, using multiple datasets, we examined how much we can improve classification accuracy by extending the explanatory variables beyond the standard scoring for sentiment, readability, positivity, negativity, risk, litigiousness, and polarity used in the finance literature. We verify that these methods deliver a high baseline level of accuracy. By using full text, we are able to inject non-numerical, forward-looking information, and the improvement in classification performance is material. For the financial news dataset, there is a 20% improvement in accuracy. For a different experiment on SEC 10-K/Q and 8-K filings, the improvement over just using return series is 5%. In general, adding text to tabular data improves classification accuracy on news and return data.

Other experiments show that the use of language models adds context and provides an additional 7% improvement in classification accuracy. This suggests that context matters in classifying financial text. We also pretrained a RoBERTa-Fin2 large model on 10 years of SEC filings and showed that it provides a better model than standard transformer-based BERT, DistilBERT, and RoBERTa models. Therefore, pretraining on finance-specific text is also valuable.

We also attempted to handle long documents via summarization, which yields no material improvement in accuracy. More work needs to be done to determine the best methods for long documents, which are ubiquitous in finance. Further work will examine models (e.g., Zaheer et al. 2020) that tackle the long-document problem. The experiments here are extensible to a fuller examination using long-document methods for classification of regulatory filings, earnings call transcripts, analyst reports, and so on.

REFERENCES

Antweiler, W., and, M. Z. Frank. 2004. “Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards.” *The Journal of Finance* 59 (3): 1259–1294.

Araci, D. “FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models.” *arXiv* 1908.10063, 2019.

Bach, M. P., Z. Krstic, S. Seljan, and L. Turulja. 2019. “Text Mining for Big Data Analysis in Financial Sector: A Literature Review.” *Sustainability* 11 (5): 1–27.

Balyuk, T., N. Prabhala, and M. Puri. “On the Costs of Being Public and Government Aid: Evidence from the Paycheck Protection Program.” Working paper, Johns Hopkins University, 2020.

Beltagy, I., K. Lo, and A. Cohan. “SciBERT: A Pretrained Language Model for Scientific Text.” *arXiv* 1903.10676, 2019.

Bodnaruk, A., T. Loughran, and B. McDonald. 2015. “Using 10-K Text to Gauge Financial Constraints.” *Journal of Financial and Quantitative Analysis* 50, no. 4 (August): 623–646.

Bonsall, S. B., A. J. Leone, B. P. Miller, and K. Rennekamp. 2017. “A Plain English Measure of Financial Reporting Readability.” *Journal of Accounting and Economics* 63, no. 2 (April): 329–357.

Brown, S., and J. W. Tucker. 2011. “Large-Sample Evidence on Firms’ Year-over-Year MD&A Modifications.” *Journal of Accounting Research* 49 (2): 309–346.

Bushee, B. J., I. D. Gow, and D. J. Taylor. 2018. “Linguistic Complexity in Firm Disclosures: Obfuscation or Information?” *Journal of Accounting Research* 56 (1): 85–121.

Chebonenko, T., L. Gu, and D. Muravyev. “Text Sentiment’s Ability to Capture Information: Evidence from Earnings Calls.” Paper 2352524, SSRN, March 2018.

Child, R., S. Gray, A. Radford, and I. Sutskever. “Generating Long Sequences with Sparse Transformers.” *arXiv* 1904.10509, 2019.

Cohen, L., C. Malloy, and Q. Nguyen. 2020. “Lazy Prices.” *The Journal of Finance* 75 (3): 1371–1415.

Cong, L. W., T. Liang, B. Yang, and X. Zhang. “Analyzing Textual Information at Scale.” Paper 3449822, SSRN, November 2019.

Cong, L. W., T. Liang, and X. Zhang. “Textual Factors: A Scalable, Interpretable, and Data-Driven Approach to Analyzing Unstructured Information.” Paper 3307057, SSRN, September 2019.

Das, S. R. 2014. “Text and Context: Language Analytics in Finance.” *Foundations and Trends in Finance* 8 (3): 145–261.

Das, S. R., and M. Y. Chen. 2007. “Yahoo! For Amazon: Sentiment Extraction from Small Talk on the Web.” *Management Science* 53 (9): 1375–1388.

Desola, V., K. Hanna, and P. Nonis. “FinBERT: Pre-Trained Model on SEC Filings for Financial Natural Language Tasks.” 2019, UC Berkeley, Working Paper 266, DOI: 10.13140/RG.2.2.19153.89442.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” version 2, *arXiv* 1810.04805, 2019.

Erickson, N., J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola. “AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data.” *arXiv* 2003.06505, 2020.

Ertugrul, M., J. Lei, J. Qiu, and C. Wan. 2017. “Annual Report Readability, Tone Ambiguity, and the Cost of Borrowing.” *Journal of Financial and Quantitative Analysis* 52, no. 2 (April): 811–836.

Gao, M., and J. Huang. 2020. “Informing the Market: The Effect of Modern Information Technologies on Information Production.” *The Review of Financial Studies* 33, no. 4 (April): 1367–1411.

Gentzkow, M., B. Kelly, and M. Taddy. 2019. “Text as Data.” *Journal of Economic Literature* 57, no. 3 (September): 535–574.

Hafez, P., M. Kangrga, J. Guerrero-Colon, F. Gomez, and R. Matas. “Capturing Alpha from Your Own Digital Content.” 2020a, www.ravenpack.com.

Hafez, P., R. Matas, F. Gomez, M. Kangrga, B. Skorodumov, and A. Liu. “RavenPack News Sentiment Data Outperforms During Coronavirus Crisis.” 2020b, www.ravenpack.com.

Hoberg, G., and G. Phillips. 2016. “Text-Based Network Industries and Endogenous Product Differentiation.” *Journal of Political Economy* 124, no. 5 (October): 1423–1465.

Huang, K., J. Altosaar, and R. Ranganath. “ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission.” *arXiv* 1904.05342, 2019.

Jegadeesh, N., and D. Wu. 2013. “Word Power: A New Approach for Content Analysis.” *Journal of Financial Economics* 110, no. 3 (December): 712–729.

Kearney, C., and S. Liu. 2014. “Textual Sentiment in Finance: A Survey of Methods and Models.” *International Review of Financial Analysis* 33 (May): 171–185.

Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. 2019. "BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining." *Bioinformatics* 36 (4): 1234–1240.

Lee, J.-S., and J. Hsiang. "PatentBERT: Patent Classification with Fine-Tuning a Pre-Trained BERT Model." *arXiv* 1906.02124, 2019.

Li, F. 2010a. "The Information Content of Forward-Looking Statements in Corporate Filings—A Naïve Bayesian Machine Learning Approach." *Journal of Accounting Research* 48 (5): 1049–1102.

——. 2010b. "Textual Analysis of Corporate Disclosures: A Survey of the Literature." *Journal of Accounting Literature* 29: 143–165.

Li, J., and X. Zhao. "Complexity and Information Content of Financial Disclosures: Evidence from Evolution of Uncertainty Following 10-K Filings." 2014, https://ssrn.com/abstract=2516622.

Liu, L., K. Liu, Z. Cong, J. Zhao, Y. Ji, and J. He. 2018. "Long Length Document Classification by Local Convolutional Feature Aggregation." *Algorithms* 11, no. 8 (August): 109.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *arXiv* 1907.11692, 2019.

Loughran, T., and B. McDonald. 2011. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks." *The Journal of Finance* 66 (1): 35–65.

——. 2013. "IPO First-Day Returns, Offer Price Revisions, Volatility, and Form S-1 Language." *Journal of Financial Economics* 109 (2): 307–326.

——. 2014. "Measuring Readability in Financial Disclosures." *The Journal of Finance* 69 (4): 1643–1671.

——. 2015. "The Use of Word Lists in Textual Analysis." *Journal of Behavioral Finance* 16, no. 1 (January): 1–11.

——. 2016. "Textual Analysis in Accounting and Finance: A Survey." *Journal of Accounting Research* 54 (4). 1187–1230.

——. 2020. "Textual Analysis in Finance." *SSRN* 3470272, June 2020.

Loughran, T., B. McDonald, and H. Yun. 2009. "A Wolf in Sheep's Clothing: The Use of Ethics-Related Terms in 10-K Reports." *Journal of Business Ethics* 89, no. 1 (May): 39–49.

Malkiel, B. G. A Random Walk down Wall Street: The Time-Tested Strategy for Successful Investing. New York: W.W. Norton, 2003.

Malo, P., A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. "Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts." *Journal of the Association for Information Science and Technology* 65, no. 4 (April): 782–796.

Matthews, B. W. 1975. "Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme." *Biochimica et Biophysica Acta (BBA)—Protein Structure* 405, no.2 (October): 442–451.

Pappagari, R., P. Żelasko, J. Villalba, Y. Carmiel, and N. Dehak. "Hierarchical Transformers for Long Document Classification." *arXiv* 1910.10781, 2019.

Price, S. M., J. S. Doran, D. R. Peterson, and B. A. Bliss. 2012. "Earnings Conference Calls and Stock Returns: The Incremental Informativeness of Textual Tone." *Journal of Banking & Finance* 36, no. 4 (April): 992–1011.

Routledge, B. R. 2019. "Machine Learning and Asset Allocation." *Financial Management* 48 (4): 1069–1094.

Sanh, V., L. Debut, J. Chaumond, and T. Wolf. “DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter.” *arXiv* 1910.01108, 2020.

Tetlock, P. C., M. Saar-Tsechansky, and S. Macskassy. 2008. “More Than Words: Quantifying Language to Measure Firms’ Fundamentals.” *The Journal of Finance* 63 (3): 1437–1467.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. “Attention Is All You Need.” *arXiv* 1706.03762, 2017.

Wan, L., G. Papageorgiou, M. Seddon, and M. Bernardoni. “Long-Length Legal Document Classification.” *arXiv* 1912.06905, 2019.

Yang, Y., M. C. S. Uy, and A. Huang. “FinBERT: A Pretrained Language Model for Financial Communications.” *arXiv* 2006.08097, 2020.

Zaheer, M., G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. “Big Bird: Transformers for Longer Sequences.” *arXiv* 2007.14062, 2020.

Zheng, S., H. Lin, S. Zha, and M. Li. “Accelerated Large Batch Optimization of BERT Pretraining in 54 Minutes.” *arXiv* 2006.13484, 2020.