Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

How Much Can Machines Learn Finance from Chinese Text Data?

Yang Zhou, Jianqing Fan, Lirong Xue

To cite this article:
Yang Zhou, Jianqing Fan, Lirong Xue (2024) How Much Can Machines Learn Finance from Chinese Text Data?. Management Science 70(12):8962-8987. https://doi.org/10.1287/mnsc.2022.01468

https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Vol. 70, No. 12, December 2024, pp. 8962–8987
ISSN 0025-1909 (print), ISSN 1526-5501 (online)

How Much Can Machines Learn Finance from Chinese Text Data?

Yang Zhou,[a,b] **Jianqing Fan,**[c,d,e,*] **Lirong Xue**[d]

[a] Institute for Big Data, Fudan University, Shanghai 200433, China; [b] MOE Laboratory for National Development and Intelligent Governance, Fudan University, Shanghai 200433, China; [c] International School of Economics and Management, Capital University of Economics and Business, Beijing 100070, China; [d] Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey 08544; [e] School of Data Science, Fudan University, Shanghai 200433, China

*Corresponding author

Contact: yangzhou@fudan.edu.cn, https://orcid.org/0000-0003-2698-6077 (YZ); jqfan@princeton.edu, https://orcid.org/0000-0003-3250-7677 (JF); lirongx.pu@gmail.com (LX)

Received: February 16, 2021
Revised: March 29, 2023
Accepted: July 18, 2023
Published Online in Articles in Advance: March 18, 2024

https://doi.org/10.1287/mnsc.2022.01468

Abstract. How much can we learn finance directly from text data? This paper presents a new framework for learning textual data based on the factor augmentation model and sparsity regularization, called the factor-augmented regularized model for prediction (FarmPredict), to let machines learn financial returns directly from news. FarmPredict allows the model itself to extract information directly from articles without predefined information, such as dictionaries or pretrained models as in most studies. Using unsupervised learned factors to augment the predictors would benefit our method with a "double-robust" feature: that the machine would learn to balance between individual words or text factors/topics. It also avoids the information loss of factor regression in dimensionality reduction. We apply our model to the Chinese stock market with a large proportion of retail investors by using Chinese news data to predict financial returns. We show that positive sentiments scored by our FarmPredict approach from news generate on average 83 basic points (bps) stock daily excess returns, and negative news has an adverse impact of 26 bps on the days of news announcements, where both effects can last for a few days. This asymmetric effect aligns well with the short-sale constraints in the Chinese equity market. The result shows that the machine-learned prediction does provide sizeable predictive power with an annualized return of 54% at most with a simple investment strategy. Compared with other statistical and machine learning methods, FarmPredict significantly outperforms them on model prediction and portfolio performance. Our study demonstrates the far-reaching potential of using machines to learn text data.

History: Accepted by Kay Giesecke, finance.
Funding: This study was supported by the National Natural Science Foundation of China [Grants 71991471, 71991470, and 72204049], the National Key Research and Development Program [Grant 2020YFA0608604], the Shanghai Pujiang Scholar Project [Grant 21PJC010], the Shanghai Science Project [Grant 23692119300], and the China Postdoctoral Science Project [Grants 2019M650076 and 2020T130107].
Supplemental Material: The online appendix and data files are available at https://doi.org/10.1287/mnsc.2022.01468.

Keywords: machine learning • FarmPredict • factor model • sparse regression • textual analysis

1. Introduction

Text data, as the most common tool for records and communications, play a critical role in social science studies as a complement to traditional structured data. Because text data from media, news, and reports can reflect the attitudes of agents in the economy, such as through comments, perspectives, objectives, and sentiments, it is useful to apply text data to financial studies (Gu et al. 2020). A common method for this unstructured text data is to transform it into a structured frame and then conduct analytical processes, such as word screening, semantics learning, and "sentiment" measuring.[1] This "sentiment" measure can be used to predict asset prices or returns in equity markets as an effective instrument for portfolio choice or asset pricing analysis (Sun et al. 2016, Gao et al. 2020). With developments in data science and modern computation power, it is possible to automatically extract such information from encoded text data by statistical machine learning methodologies.

Traditional studies typically count the number of particular words in the overlap of the document and a

predefined dictionary. Loughran and McDonald (2016) introduced the most widely used dictionaries in a review, including dictionaries proposed in Henry (1973), Harvard's General Inquirer Word List, Diction Optimism and Pessimism Word Lists, and the widely applied list from Loughran and McDonald (2011). It is proven that the dictionary approach can provide a significant correlation between sentiments and stock returns,[2] but it can also be a double-edged sword. Researchers can easily replicate or extrapolate the analysis with public dictionaries, whereas the results highly rely on the dictionary, which can be easily biased because of subjective human experience. Therefore, recent studies are focusing on dictionary building based on a machine learning framework (Du et al. 2022).

Regarding these limitations, studies tried to apply machine learning methods to text data. One common issue in textual analysis is how to extract useful information instead of noise from high-dimensional but sparse predictors. Naturally, such an issue can be treated as a dimension-reduction problem by either selecting key variables (words or phrases) or clustering/grouping (textual factors). Most studies differ in details and application of these methodologies, say how to use "machines" to "learn" text data.

One vein of financial textual analysis is word selection either by text regression with a penalty or by generative (topic) models based on the path of generating languages via machine learning algorithms (Gentzkow et al. 2019a). As an early pioneering study, Antweiler and Frank (2004) collected information from 45 companies and used a naive Bayes model to predict their stock prices and returns. Taddy (2013) proposed the multinomial inverse regression (MNIR) model for dimension reduction, where predictors (words or phrases) were represented as draws from a multinomial distribution. Jegadeesh and Wu (2013) conducted a text regression to assign weights to words based on market returns. With a similar research framework, Manela and Moreira (2017) used supported vector machines, a nonlinear penalized regression approach to screen words for volatility prediction in the financial market.

Based on the language generation process, generative topic models were proposed, mainly based on the latent Dirichlet allocation (LDA) (Blei et al. 2003). LDA not only focuses on the weight or coefficient of a single word but regards the document as the result of the generative process of one topic, which shares the thought in Taddy (2013). Following this spirit, Gentzkow et al. (2019b) measured trends in the partisanship of congressional speech, and Ke et al. (2019) proposed a supervised sentiment model to predict returns in stock markets. Despite the advantages of topic models, they still rely heavily on prior knowledge and statistical assumptions, especially in the model set. This close reliance limits the adaptiveness of the textual model as it may only provide ad hoc results that cannot be replicated or achieve the same accuracy in other sectors or markets. Moreover, semantic information is not the only dimension of a document (Calomiris and Mamaysky 2019), and the holistic application to a document would provide more information on forecasting and prediction. Therefore, even though previous models have demonstrated fair predictive capacity and returns in the stock market, it is still unclear how much machines can learn from this comprehensive text data.

The recent development of natural language processing has provided an alternative way of dimension reduction in text data by clustering/grouping several words into one factor/topic. For example, Bidirectional Encoder Representations from Transformers (BERT) or word2vec methods transformed words to vectors (Devlin et al. 2019), which allows us to view each word in a high-dimensional space, and hence, to calculate the distance between them for clustering. For example, Cong et al. (2019) provided a word2vec-based textual factorization framework for textual analysis in social science studies.

From all these points, this paper introduces a novel factor-augmented regularized model for prediction (FarmPredict) on stock returns by extracting the hidden topics (factors) from a particular article for predictor augmentation. Because FarmPredict does not rely on preobtained information, it is a more general analytical framework, providing a highly adaptive modeling process for studying text data.

FarmPredict consists of three steps. The first step is to learn hidden features from high-dimensional articles without supervision. To do this, we convert articles into vectors of hidden components consisting of multiple factors and idiosyncratic residuals via principal component analysis (PCA). The number of hidden factors is learned by the adjusted eigenvalue thresholding method (Fan et al. 2020a). It is a pure unsupervised learning process without forced intervention from prior assumptions. We then screen the idiosyncratic variables by their correlations with our learning target, the corresponding beta-adjusted returns,[3] conditional on factors. This step is optional but helps us reduce dimensionality to a more manageable level. Finally, we apply a simple least absolute shrinkage and selection operator (LASSO) method (or other machine learning algorithms) to predict asset prices using hidden factors and screened idiosyncratic components. Therefore, as an analytical framework, FarmPredict is highly flexible in data construction, the screening process, and prediction model selection.[4]

Our study gathers financial news from *Sina Finance*, one of the major news hubs for Chinese equity markets. The website publishes over 500 news stories daily and offers timely and comprehensive coverage of all the popular financial news in Chinese. We used crawling to download publicly available news web pages from its website and extracted related time, text, and stock

information for our data. The text is segmented with a hidden Markov model and paired with returns with corresponding code and time. Each article is paired with its effective beta-adjusted returns for model training. We fitted FarmPredict on these data and predicted corresponding returns from 2015 to 2019. (The data can be accessed at https://www.icloud.com/iclouddrive/058xOEPIXtI_vB0qgTFRDeBcw#codes.data_wordomit2.)

We then validated the sentiment scores from FarmPredict via multiple approaches. First, we examined the meanings of major sentiment-charged words selected by our model and demonstrated that FarmPredict could capture more interactive and abnormal information. The panel regression also demonstrated that FarmPredict can learn specific information about target stocks, resulting in a significant correlation with the beta-adjusted returns of targeted stocks. We also treated the news in this paper as "events" and estimated the pattern of stock returns based on an event study. It revealed the potential mechanism of how unexpected news can affect the financial markets in China. The results showed that the beta-adjusted returns started to increase about seven days before the occurrence of positive news, whereas no such result was observed for negative news. This asymmetric effect of impact aligns well with the short-sale constraints and supervisions in the Chinese equity market, which make the leak or anticipation of negative news harder to react to (Nagel 2005, Chen et al. 2019). After impact peaking on the news arrival day, with an average of 83 basic points (bps) on positive news sentiments and 26 bps on negative ones, the (positive/negative) impact of news arrivals would last for a few days. A placebo test lends further support to this result; thus, this leads to investment opportunities.

We also tested our machine learning methodology in terms of financial investments. We built daily portfolios based on sentiment scores and recorded their returns. Despite the high trading cost in the Chinese stock market (about 13 bps per trading), the annualized percentage return (APR) of the daily portfolio after high transaction costs and daily price limits still reached 54% (Sharpe ratio (SR): 4.30) for the equally weighted (EW) portfolio and 9% (Sharpe ratio: 0.55) for the value-weighted (VW) one[5] during the test period of 2015–2019, significantly exceeding other models. We further analyzed the portfolio's risk exposure and return from alpha or different components to reveal the mechanism of such performances.

We then discussed the model in more detail. First, we decomposed the model and evaluated the contribution of each component separately. Then, we presented the contents of the factors and summarized them into potential topics. FarmPredict is compared with other state-of-the-art statistical and machine learning models, such as MNIR, sentiment extraction via screening and topic modeling (SESTM), textual factor, BERT, neural networks, and traditional momentum strategy. To further verify the robustness of FarmPredict, we tested the model's sensitivity in terms of various transformations of input and output, choice of factors, screening level, number of stocks in constructed portfolios, and number of news inputs.[6] The stable results demonstrate the robustness of FarmPredict.

Our model has important implications for understanding how much financial information machines can learn from text data as well as the return prediction and realization by text-based sentiment studied by a rich set of papers. First, our FarmPredict starts with an unsupervised factor extraction of words, and all parameters are determined in the training process. Therefore, FarmPredict does not rely on any prior assumptions or experiences but only conducted a data-driven process. This choice provides a significant benefit to text modeling; let machines learn the meaning of the key components of the text without supervision by human experience. The sole data-driven process also leads to high flexibility, suitability, and robustness of our model on text data analysis because hidden factors and features can be revealed by machine learning without any intervention from predefined knowledge, hence avoiding potential subjective bias.

Second, the FarmPredict is not only a model but an analytical framework of machine learning for high-dimensional data, which are text data in this paper. By transforming the original data into the latent factors and idiosyncratic components, FarmPredict effectively converts high-dimensional data with highly correlated covariates into weakly correlated ones in an unsupervised way. Hence, FarmPredict could solve the statistical obstacle of multicollinearity. The subsequent marginal screening performs an efficient dimensional reduction and selects the most related and predictive words. It is worth noting that the screening process in FarmPredict is conditional on hidden factors being learned from all elements (words) in the data, resulting in the use of all information without supervision. Thanks to all these features, the framework of FarmPredict is very flexible in learning factors, idiosyncratic components, methods for screening, and selection of linear or nonlinear models for prediction.

Differing from the dimension-reduction processes by word selection or clustering method in previous papers, FarmPredict used unsupervised learned factors to augment the predictors. Covering both factors and residuals would also benefit FarmPredict with a "double-robust" feature; the model would "automatically" balance between word selection and clustering. For example, if the return can be perfectly predicted by word selection, FarmPredict would result in zero hidden factors and "collapse" into an LASSO model for an optimal estimation.

Third, most studies are conducted under a language environment in English and relatively developed financial markets, whereas very few studies focus on other languages and developing or emerging markets (Calomiris and Mamaysky 2019). This paper showed the possibility of applications of machine learning techniques in languages other than English and developing markets.[7] As the second-largest economy in the world, the equity market in China is too big to ignore. Compared with the structure of market participants in the United States, there are significantly more individual retailers than institutional wholesalers in China, leading to higher uncertainty and irrationality. Moreover, as a developing market, Chinese financial supervision imposes stricter restrictions to regulate trade and stabilize financial markets, such as imposing limits on daily equity price movements and short actions(Chen et al. 2019). It remains unclear how text data will perform in such conditions.

Finally, simply by longing the high-score stocks and shorting the low-score ones, our portfolio-building strategy based on machine-learned sentiment scores can achieve significantly outperformed returns. By comparing the returns before and after news occurs, this paper also provides information transmission, particularly in the financial sector in China. Our research completes the vein of literature on textual analysis, expands the depth of statistical machine learning techniques in financial studies, and sheds light on the rich application of machine learning to social science topics.

The remainder of the paper is organized as follows. Section 2 introduces FarmPredict. Section 3 describes our data and the detailed analysis process. Section 4 provides empirical results to validate FarmPredict. Section 5 discusses the model and compares FarmPredict with other methods. Section 6 concludes the paper.

2. Methods

This section discusses the framework of using machines to learn text data. We first summarize the framework and notations and then introduce details of FarmPredict. Variations of the FarmPredict framework then follow.

2.1. Problem Setup

We use the word-level statistics as a summary of each of the n articles. Let $\mathbf{D}$ be the set of all possible Chinese words in our data of n articles and $\mathbf{d}_i \in \mathbb{N}^{|\mathbf{D}|}$ be the vector of word counts of every word in the ith article, with $d_{i,k}$ being the number of times the kth word appears in the article.

Article i is associated with a target outcome or response Y_i, which in this paper, is the beta-adjusted return of the corresponding stock on the day the news was published. The data are very high dimensional and appear sparsely in each article, especially in Chinese. In our data set of 914,000 articles, there are 1,181,000 distinctive words[8] in the entire set $\mathbf{D}$, whereas only 71,000 words appear in at least 50 articles in the data. Following previous papers, we assume that the target responses $\{Y_i\}$ are mainly affected by a relatively small subset of words, which are defined as sentiment-charged words. Such an assumption also helps us reduce the dimensionality of the data to a reasonable level.

Hence, all the words can be divided into two disjoint categories (the set of sentiment-charged words $\mathbf{S}$ and the set of sentiment-neutral words $\mathbf{N}$ so that $\mathbf{D} = \mathbf{S} \cup \mathbf{N}$), whereas the sentiment score of an article is mainly associated with its sentiment-charged words.

2.2. FarmPredict

Most traditional textual analyses, like topic models or dictionary-based methods, are conducted with several restrictions, such as the determination of topics and the overlapping of the information. Such a condition would result in inflexibility and possible inaccurate estimation. A natural question is then if we can learn the sentiments directly from high-dimensional regression as sentiment prediction in finance is fundamentally a regression problem. Here, we propose a direct regression framework called FarmPredict.

FarmPredict uses both factors and idiosyncratic residuals to enhance the prediction. When no factors are selected, it reduces to the ordinary LASSO. Hence, FarmPredict possesses a "double-robust" feature; it "automatically" balances between word selection and word clustering. For example, if the return can be perfectly predicted by word selection, FarmPredict selects no latent factors and uses an LASSO model for optimal prediction. At the same time, FarmPredict overcomes the information loss by using only principal components in the dimensionality reduction and alleviates model selection inconsistency by penalized methods, such as LASSO, for high-correlated covariates (Fan et al. 2020b).

2.2.1. Selecting Frequent Words. Of the over 1.1 million distinct words (and phrases) of our data set, most of them rarely occur. As such, we begin by filtering out these infrequent words that only appear in a small fraction of articles. These words are also hardly useful as they are unlikely to appear in new articles to be scored. The screening also helps us narrow our focus to a reasonably comprehensive set of words $\mathbf{D}^{\text{freq}}$, around 10,000 or so.

Let k_j be the number of articles that contain the word j. For a threshold κ, we keep the vocabulary

$$\mathbf{D}^{\text{freq}} = \{j\text{th word in } \mathbf{D} : k_j \geq \kappa\}. \tag{2.1}$$

The threshold κ will be tuned as a hyperparameter to strike a balance between the comprehensiveness of $\mathbf{D}^{\text{freq}}$ and the noises introduced by infrequent words.

2.2.2. Factor Modeling. Let $\mathbf{X}_i$ be the feature vector in which $X_{i,j}$ is the feature of word $j \in \mathbf{D}^{\text{freq}}$ in the ith article. It can be the original word counts or simply $\{0, 1\}$, indicating the absence or presence of the word j in the ith article. The dependence among words is assumed to be driven by some latent factors. Namely, $\mathbf{X}_i$ follows an approximate factor model

$$\mathbf{X}_i = \mathbf{B}\mathbf{f}_i + \mathbf{u}_i, \qquad i = 1, \ldots, n, \tag{2.2}$$

where $\mathbf{f}_i \in \mathbb{R}^k$ is the vector of k latent factors, $\mathbf{B}$ is the factor loading matrix, and $\mathbf{u}_i \in \mathbb{R}^{|\mathbf{D}^{\text{freq}}|}$ is a vector of idiosyncratic components that cannot be explained by (or uncorrelated with) $\mathbf{f}_i$. Putting the factor model in the matrix form, we have

$$\mathbf{X} = \mathbf{F}\mathbf{B}^T + \mathbf{U},$$

where $\mathbf{X}$ and $\mathbf{U}$ are $n \times |\mathbf{D}^{\text{freq}}|$ matrices of data and idiosyncratic components and $\mathbf{F}$ is $n \times k$ of latent factors. Here, only $\mathbf{X}$ is observable, and $\mathbf{F}, \mathbf{B}, \mathbf{U}$ will be estimated by PCA.

The factors can be understood similarly to themes or topics of an article, and the factor loading matrix $\mathbf{B}$ extracts the mix of these factors (topics) from an article. For example, macroeconomy news and fund performance articles might each have their own distinct vocabularies, represented as the vector difference in the loading matrix $\mathbf{B}$, and hence, the corresponding stock return is influenced by the combination of factors.

The factor model disentangles correlated features in $\mathbf{X}_i$ by decomposing them into factors $\mathbf{f}_i$ and idiosyncratic components $\mathbf{u}_i$. Suppose that we would like to use $\mathbf{X}_i$ to predict the associated return outcome Y_i. Following a similar idea in Fan et al. (2020b), we use latent $\mathbf{f}_i$ and $\mathbf{u}_i$ as the predictor and build the model

$$Y_i = a + \mathbf{b}^T\mathbf{f}_i + \boldsymbol{\beta}^T\mathbf{u}_i + \epsilon_i, \tag{2.3}$$

where ϵ_i is the idiosyncratic noise. This model is broader than the linear model in $\mathbf{X}_i$, augmenting the predictors using latent factors $\mathbf{f}_i$, and the variables in Equation (2.3) are less correlated. We will additionally impose a sparsity constraint on $\boldsymbol{\beta}$ and $\mathbf{b}$ as most words do not carry signals on an article's sentiments or stock returns.

Note that the linear space spanned by $\mathbf{X}_i$ and $\mathbf{f}_i$ is the same as that spanned by $\mathbf{u}_i$ and $\mathbf{f}_i$. Therefore, we expand the model in the useful directions using the latent factors $\mathbf{f}_i$. The novelty of the method is that the factors can be learned from the original data $\mathbf{X}_i$ but can also be learned from different variables, such as bivariate interactions of $\mathbf{X}_i$, or even from augmented data that include the firm's characteristics. This significantly increases the versatility of our approach.

2.2.3. Learning Factors and Idiosyncratic Components. For a given number of factors k, we fit the approximate factor model (2.2) via least squares, resulting in principal component analysis. The solution[9] is to estimate latent factor $\widehat{\mathbf{F}} = \sqrt{n}$ times the eigenvectors of the largest k eigenvalues of matrix $\mathbf{X}\mathbf{X}^T$, $\widehat{\mathbf{B}} = \mathbf{X}^T\widehat{\mathbf{F}}/n$, and $\widehat{\mathbf{U}} = \mathbf{X} - \widehat{\mathbf{F}}\widehat{\mathbf{B}}^T$.

There are several data-driven methods for selecting the number of factors k. See Fan et al. (2020c) and the references therein. Here, we use the adjusted eigenvalue thresholding (Fan et al. 2020a). The method takes into account the heterogeneous scales of observed variables and estimates the number of factors via thresholding on bias-corrected estimators of eigenvalues of the correlation matrix. Specifically, k is estimated as the number of corrected eigenvalues that are statistically larger than one:

$$\hat{k} = \max\left\{j < |\mathbf{D}^{\text{freq}}| : \hat{\lambda}_j^C > 1 + C\sqrt{|\mathbf{D}^{\text{freq}}|/(n-1)}\right\}, \tag{2.4}$$

where $\hat{\lambda}_j^C$ is the bias-corrected estimator of the jth-largest eigenvalue of the correlation matrix of the data matrix $\mathbf{X}$.[10]

2.2.4. Learning Conditional Sentiment-Charged Words S. With learned factors in place, we can further screen down the predictive words (sentiment-charged words) using conditional correlation screening. Let $\widehat{\mathbf{Y}}_u$ be the residual vector of $\mathbf{Y}$ after fitting a linear regression of $\mathbf{Y}$ on $\widehat{\mathbf{F}}$ with intercepts. This takes out the part of $\mathbf{Y}$ that can be explained by the factors. We seek components of $\widehat{\mathbf{U}}$ to further predict $\widehat{\mathbf{Y}}_u$.

Conditional screening is to seek words that have a high correlation with $\mathbf{Y}_u$ (Fan and Lv 2008): more precisely, the correlation between $\widehat{\mathbf{Y}}_u$ and the idiosyncratic component $\widehat{\mathbf{U}}_j$ for word j, which is the jth column of $\widehat{\mathbf{U}}$. This correlation is the partial correlation between $\mathbf{Y}$ and the feature vector associated with word j, conditioning on the latent factors $\mathbf{F}$. Given a threshold α, the conditional sentiment-charged words are defined by

$$\widehat{\mathbf{S}} = \{j : |\text{corr}(\widehat{\mathbf{U}}_j, \widehat{\mathbf{Y}_u})| > \alpha\} \cap \{j : k_j \geq \kappa\}. \tag{2.5}$$

It is worth noting that this step is optional (corresponding to $\alpha = 0$) but helps us speed up computation.

2.2.5. Fitting FarmPredict. With the hyperparameters in place, we can train our regression model flexibly with statistical and machine learning models. In the paper, among the conditional sentiment-charged words, FarmPredict solves the penalized least squares:

$$\hat{a}, \widehat{\mathbf{b}}, \widehat{\boldsymbol{\beta}} = \underset{a, \mathbf{b}, \boldsymbol{\beta}}{\arg\min} \left\{ \frac{1}{n}\sum_i (Y_i - a - \mathbf{b}^T\mathbf{f}_i - \boldsymbol{\beta}^T\mathbf{u}_{i,\hat{\mathbf{S}}})^2 + \lambda\|\boldsymbol{\beta}\|_1 + \lambda\|\mathbf{b}\|_1 \right\}, \tag{2.6}$$

where $\mathbf{u}_{i,\hat{\mathbf{S}}}$ is the components of $\mathbf{u}_i$ restricted to the

sentiment-charged words $\widehat{\mathbf{S}}$. The penalty parameter λ, which will be chosen by the crossvalidation, controls the models' bias-variance trade-off and also, the sparsity of $\widehat{\boldsymbol{\beta}}$ and $\hat{b}$. This further reduces sentiment-charged words. Note that the number of factors is usually small and that a viable alternative is not to penalize the coefficients $\mathbf{b}$.

The LASSO penalty in Equation (2.6) can also be changed to other functions, such as smoothly clipped absolute deviation and elastic net, among others (Fan et al. 2020c, Nagel 2021).

2.2.6. Scoring New Articles. Scoring a new article consists of two steps. For a given new feature vector $\mathbf{X}_{\text{new}}$, we decompose it into factors and idiosyncratic components with a well-trained $\widehat{\mathbf{B}}$, applying the least squares to model (2.2). Hence, we could obtain the latent factor $\mathbf{f}_{\text{new}}$ as well as the idiosyncratic component $\mathbf{u}_{\text{new}}$ associated with the feature $\mathbf{X}_{\text{new}}$ as follows:[11]

$$\mathbf{f}_{\text{new}} = (\widehat{\mathbf{B}}^T\widehat{\mathbf{B}})^{-1}\widehat{\mathbf{B}}^T\mathbf{X}_{\text{new}}, \qquad \mathbf{u}_{\text{new}} = \mathbf{X}_{\text{new}} - \widehat{\mathbf{B}}\widehat{\mathbf{f}}_{\text{new}}. \tag{2.7}$$

Therefore, its sentiment score is predicted as

$$\hat{Y}_{\text{new}} = \hat{a} + \widehat{\mathbf{b}}^T\mathbf{f}_{\text{new}} + \widehat{\boldsymbol{\beta}}^T\mathbf{u}_{\text{new},\hat{\mathbf{S}}}. \tag{2.8}$$

2.3. Variations on FarmPredict

Because FarmPredict directly learned information from the data, it is highly versatile and adaptive to different tasks. First of all, the response variable Y can be raw beta-adjusted returns or dichotomous returns (positive or negative). In the latter case, one can use penalized least squares as in (2.6) or penalized logistic regression.[12]

Second, the feature vector can be the original counts or their modified version, such as the dichotomized ones (absence and presence). In the latter case, an alternative extraction of latent factors can also be obtained, and the factor loadings on the dichotomized features can be learned from least squares or logistic regression.

Finally, the linear prediction model (2.3) can be replaced by nonlinear models

$$Y_i = g(\mathbf{f}_i, \mathbf{u}_{i,\mathbf{S}}) + \epsilon_i,$$

such as neural network models (Horel and Giesecke 2020) or structured nonparametric models (Fan et al. 2020c).

In summary, FarmPredict is designed in a highly customizable way to allow for many ad hoc modifications on inputs, word screening, techniques for fitting regression functions, etc.

3. Data and Analysis

3.1. Data Collection

We used the news data from the *Sina Finance* website, one of the largest Chinese financial news websites. It publishes over a thousand Chinese stock-related news stories every day and covers most stocks in the market.

The downloading process can be viewed as searching on the internet. We started from the root of the net (main page). Nodes (web pages) in the net are connected if one has a link that points to the other, and we visited them sequentially by their distance to the root. Technically, we scrawled in a breadth-first fashion. Starting with the main page of *Sina Finance* and *Sina Caijing*, we downloaded the html file of the web page, saved it, analyzed the contents in it to get all links to other web pages, screened the links to only keep the ones inside domain finance.sina.com.cn or the domain cj.sina.com.cn, and finally, pushed the obtained links to a queue to visit later. We iteratively looped through this process for each link in the queue and let our crawler program run for several months from the end of 2019 through 2020.

In summary, we visited 6.3 million links, among which 5.8 million are valid news articles. Because of the net-like search structure of our crawling, the number of news articles we downloaded is a little random and is not exactly the same across the years.

For each web page downloaded, the published time and title are extracted from corresponding html headers. The main articles are extracted from corresponding html sections with identification as "article." For web pages without an identification, we analyzed their html structure and applied case-specific article extractors using a combination of html structures and regex expressions.

3.2. Preprocessing

We went through a series of data preprocessing steps to clean, select, and prepare the downloaded data for model fitting. We began by removing duplicated and similar articles. If two articles have the same title after removing special characters and are published on the same day, then only one will be kept in our data set. The remaining articles are then cleaned as follows.

First, all contents and titles are trimmed to Chinese characters, so all the html digits, punctuation marks, special characters, and remaining html codes are stripped away.

Then, the articles are matched with stocks. We used a combination of html and article content to find the matching stocks. We searched for the website's special stock specifier identification by regex on the entire html file to see if the page is tagged with some stocks officially by *Sina*. For pages without such a tag, we scan the article to match stock names and symbols. Articles attached to zero or more than one stock are removed.

Each remaining article is then matched with the return of its associated stock. We used beta-adjusted returns, which are calculated as the stock's returns

minus its market-induced returns as follows:

$$\text{Beta-adjusted Return}_{it} = \text{Dividend Adjusted Return}_{it} - \beta_i \cdot \text{SSEC Return}_t,$$

where the beta (β_i) of stock i is calculated by regressing its daily return on the daily returns of the Shanghai Stock Composite Index (SSEC; market return[13]) using data from 2005 to 2014.

There are several options on what kind of return and what time range of return should be used. We used the time range of *effective return*, which reflects the news' impact on the stock, covering the article's publish time. The time range is chosen carefully, so it can reflect the immediate price impact of the article. The close-to-close return covering the article's publish time is applied in this paper. For example, if an article is published at 1 p.m. inside trading hours on Tuesday, then the return from Monday market close to Tuesday market close is used. If an article is published at 6 p.m. after the market closes on Friday, then the return from the current Friday market close to next Monday market close will be used. Dividend payments and stock splits are also merged into returns to correctly reflect the stocks' actual value changes. Some stocks might not be matched with a valid return at a certain time for reasons like a trading halt, etc. We dropped the articles without matching returns.

Finally, we used Jieba[14] (Sun 2017) to divide an article's title and main text to lists of words (and phrases) based on the hidden Markov model. This method works at the single Chinese character level and labels each character as one of the four states: B (begin), M (middle), E (end), and S (single). With their existing emission and transition probability on every state and every single Chinese character, the Viterbi algorithm is used to find the most likely sequence of hidden states. Then, the text can be divided into words and phrases using the estimation results of hidden states. We chose the algorithm for its ability to deal with unknown phrases and fast speed (linear time complexity with respect to the number of characters). The number of articles after each operation is listed in Table 1.

Table 1. Number of *Sina Finance* Articles After Each Stage of Preprocessing

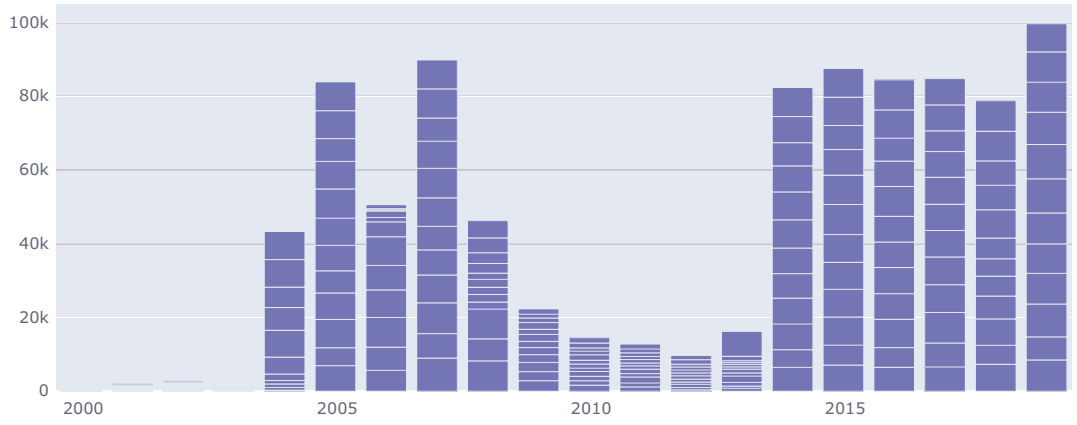
Sina Finance articles	Number of articles
All html downloaded	6,343,491
Removed nonarticles	5,880,943
Removed very similar articles	4,195,741
Removed missing date/time	4,195,726
Matched with at least one stock	2,465,127
Matched with exactly one stock	1,985,781
Matched with an effective return	1,791,364
Down sampled (≤300/days)	914,070

In the final step, we down sampled our training data to lower the computing burden[15] and balance the number of articles each day. As shown in Figure 1, our sample is not yearly balanced because of the strategy of crawling. There were over 700,000 articles downloaded in 2019, whereas only 10,000 were downloaded in 2012. We randomly down sampled the data to at most 300 articles each day. The amount of data is reduced to 914,000 articles in total and much more evenly distributed among the days.[16]

3.3. Basic Statistics

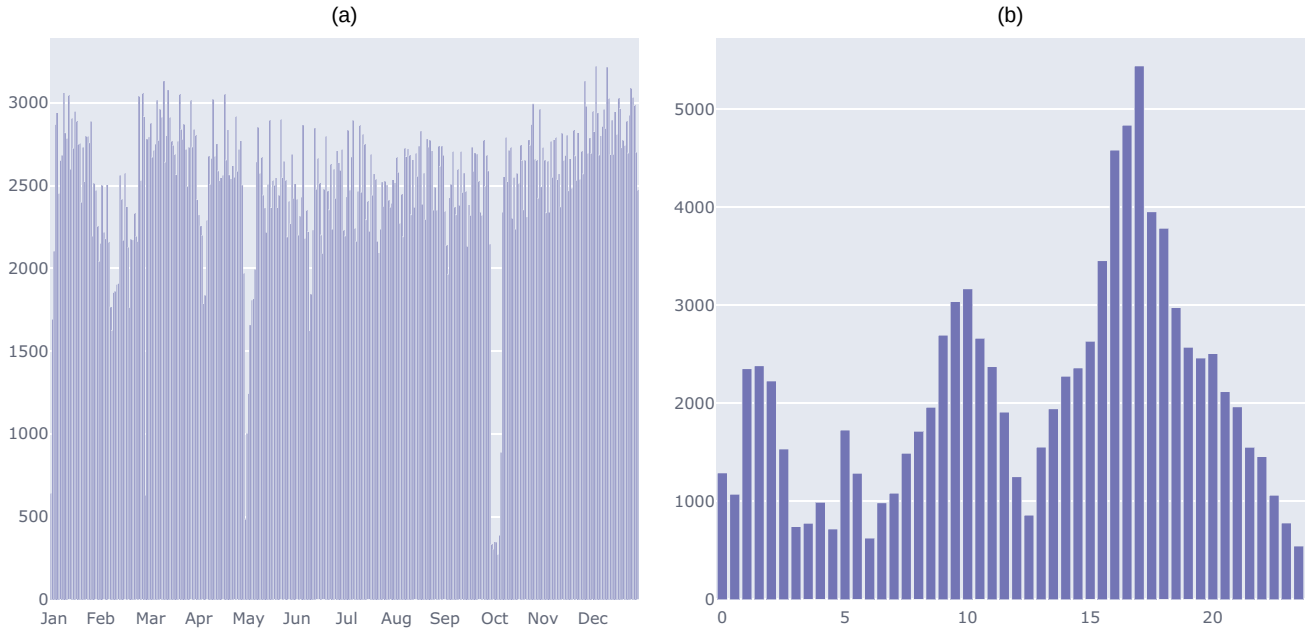
In our data set of 914,000 articles, there are 1,181,000 words (and phrases) in the entire set **D**, of which 71,000 words (and phrases) appear in at least 50 articles (0.004% of all articles). Hence, we used these basic screenings with 71,000 words and their corresponding word counts in all following models. The word count

Figure 1. (Color online) The Number of Data Points for Each Year in Our Final Data Set



Notes. The data set was down sampled so that each day has at most 300 articles. The thin white lines inside each year's bar divide data by months. Data from 2000 to 2014 are used for training and tuning, and only data from 2015 to 2019 are used for testing.

Figure 2. (Color online) The Number of Data Points Distributed on Each Day and the Number of Data Points by Time of Day in 2019



Notes. Data are largely evenly distributed across days except for the three major holidays in China. Most news is published around market open and market close. (a) Number of data points by day of year. (b) Number of data points by time of day in 2019.

matrix is highly sparse, with each article having a median of 309 words and 209 distinctive words, resulting in a median article with only 0.29% nonzero entries among 71,000 dimensional word count vectors.

We present the number of articles collected at the day-year level in Figure 2(a). The number of data points is evenly distributed across each day, except for a couple of holidays.[17] The number of data points aggregated along each half-hour window of a day is also plotted in Figure 2(b). Most news is published from the market open time around 9 a.m. to the end of the day. There is also some news published after midnight, which is mostly autogenerated news or overseas news.

More details of the data are presented in Table 2. We reported the word counts and associated returns at the single news level. In addition, we summarized the data from five years (2015–2019) of testing data by the date associated with their effective returns. The number of articles, the number of distinct stocks covered, and SSEC returns are reported. We also reported the percentage of news that is associated with a positive return.

3.4. Tuning and Testing

3.4.1. Tuning Hyperparameters. Even though most parameters can be learned in the training process, we still need to confirm some hyperparameters first, such

Table 2. Summary Statistics of Collected Data

Data	Basis	#Data	Mean	Std	Skewness	Kurtosis	10%	25%	50%	75%	90%
# words	All	914,070	680	1,077	6.5	120.6	77	152	376	781	1,440
# distinct words	All	(articles)	278	255	2.5	12.6	54	99	209	373	578
Returns	All		0.4%	5.3%	68.3	9,903.5	−3.3%	−0.9%	0.0%	1.5%	4.7%
Beta-adjusted returns	All		0.3%	5.1%	75.7	11,365.0	−2.9%	−1.1%	0.0%	1.4%	4.2%
# articles	Daily 2015–2019	1,220	356	139	2.2	5.5	268	289	308	349	549
# distinct stocks	Daily 2015–2019	(days)	250	81	1.7	3.4	184	206	231	261	370
% positive returns	Daily 2015–2019		47%	19%	0.1	−0.5	23%	34%	46%	61%	73%
SSEC returns	Daily 2015–2019		0.0%	1.5%	−1.0	6.4	−1.4%	−0.5%	0.1%	0.6%	1.6%

Notes. We summarize our data on two bases. The "all" basis looks at the entire data set and views each article as a data point. The summary statistics of each article's number of words, number of distinct words, associated effective raw returns, and beta-adjusted returns are displayed. On the "Daily 2015–2019" level, we group and summarize articles from 2015 to 2019 by their publish date. For each day, we calculate the number of articles, the number of reported distinctive stocks, SSEC return, and the proportion of articles associated with a positive return on that day. Std, standard deviation.

as the C in Equation (2.4). Tuning these hyperparameters is conducted with the data from 2000 to 2014. More specifically, we used data from 2000 to 2010 as the training set and data from 2011 to 2014 as the validation set for selecting optimal hyperparameters, which maximized the cumulative daily returns of an equally weighted portfolio.[18] The optimal combination of hyperparameters, which provided the highest cumulative returns, was fixed and used in all subsequent learning and testing. We applied these tuned hyperparameters to train other parameters and test our model's performance from 2015 to 2019 with a rolling window set, as introduced in the next section.

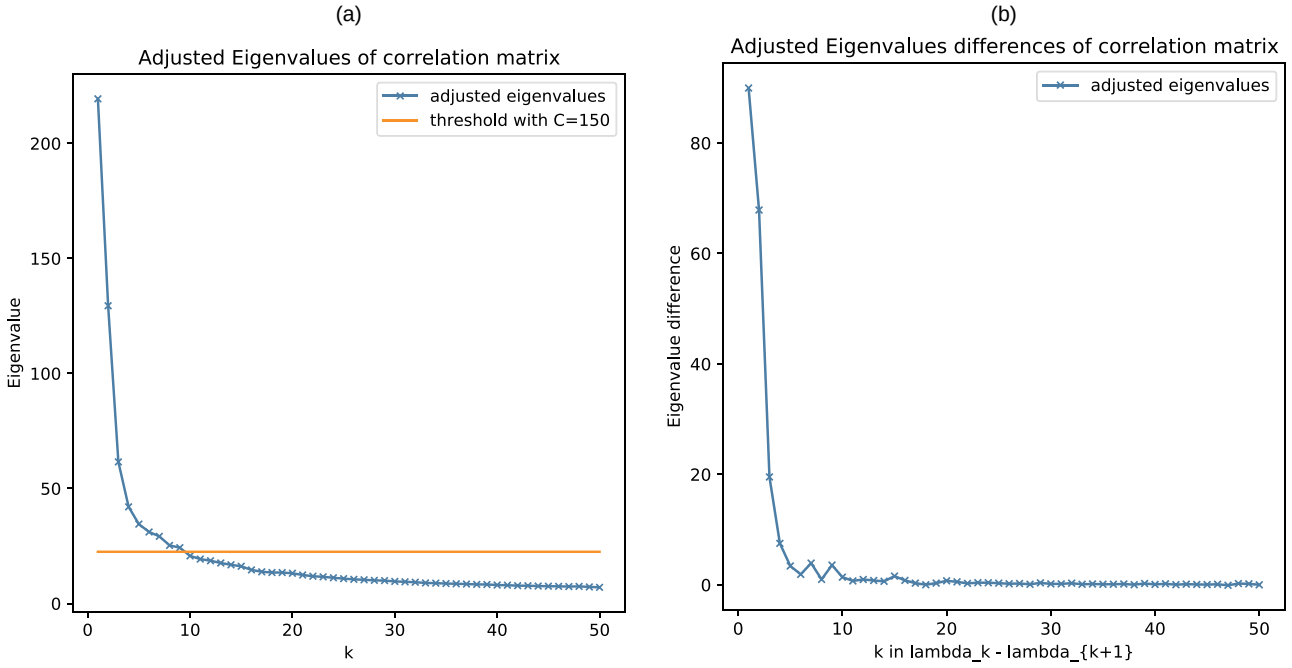
In FarmPredict, tuning starts with finding C in Equation (2.4), which controls the number of underlying factors. Figure 3 shows the screen plot and eigen differences plot of data from 2000 to 2014 using the adjusted eigenvalues of the correlation matrix of binary word counts. The figure shows that there are two stronger factors and seven relatively weaker factors. Inspired by this, we choose $C = 150$, which gives $\hat{k} = 9$ factors in the adjusted eigenvalue thresholding method (2.4). We then fix $C = 150$ through the study.[19]

With C fixed in FarmPredict, we need only to tune κ for screening frequently used words in $\mathbf{D}^{\text{freq}}$ and α for screening sentiment-charged words in $\mathbf{S}$. The tuning parameter κ is chosen from the 80%–96% quantiles of k_j's of all words, with increments of 2%. There are around 70,000 words in each 10-year training period, and the range of κ corresponds to the range of 3,000–15,000 words from $\mathbf{D}^{\text{freq}}$. The tuning parameter α is the threshold in conditional correlation screening for controlling the number of words selected into $\widehat{\mathbf{S}}$. It is chosen to ensure that the number of remaining words $|\widehat{\mathbf{S}}|$ is exactly 500, 1,000, or 2,000. A further selection of sentiment-charged words is done via penalized regression (2.6) with λ chosen by the crossvalidation.

3.4.2. Rolling Windows Test. All methods are trained and tested via rolling windows for the basis of six months. For each window, 10 years of data are used for training models, and the subsequent six months of data are then used for testing. Then, we will roll forward the entire window by six months and redo the training and testing, and we repeat with the following data. The first window is set from 2005 to 2014 for training and January to June 2015 for testing, and the test sample of the last window covers July to December 2019. In total, 10 windows are examined, and we recorded the predicted result on every trading day from 2015 to 2019.

The training and testing windows in our rolling window test are carefully chosen based on the distribution of our data. The amount of training and testing data is stable across windows. Among the 10 windows, the

Figure 3. (Color online) Top Adjusted Eigenvalues and Eigen Differences of the Correlation Matrix of Binary Word Counts



Notes. There are two major factors and seven relatively weaker factors corresponding to $C = 150$ in adjusted eigenvalue thresholding (2.4). (a) Top adjusted eigenvalues. (b) Top adjusted eigenvalue differences.

Figure 4. (Color online) Top Sentiment Words Estimated by FarmPredict

Notes. The top 50 words by their sentiment strength are selected, and their font sizes are proportional to their sentiment strengths. We selected only words in $\widehat{\mathbf{S}}$ and used their regression coefficients in $\widehat{\boldsymbol{\beta}}$ as sentiment strength.

number of training articles ranges from 428,000 to 529,000, with input words ranging from 761,000 to 863,000.

4. Results

4.1. Validation of Sentiment Scores

4.1.1. Sentiment-Charged Words. To verify our sentiment indices extracted from the context of news, we first report the top sentiment-charged words by FarmPredict. Figure 4 presents the top positive and negative words selected. We adopted the Chinese style of coloring, where red indicates positive sentiments and green indicates negative sentiments. The font size of each word is proportional to its sentiment strength in the model. In FarmPredict, we selected only words in $\widehat{\mathbf{S}}$ and used their regression coefficients in $\widehat{\boldsymbol{\beta}}$ as the sentiment strength.

Because our study focused on Chinese text data, we also present the *pinyin* for pronunciation and the translations of top positive and negative words in Table 3.

Table 3. Top Sentiment-Charged Words Chosen by FarmPredict and Their Corresponding Pinyin and English Meanings

	Positive words			Negative words		
Rank	Chinese	Pinyin	English	Chinese	Pinyin	English
1	涨停	Zhang Ting	Reach daily upper limit	跌停	Die Ting	Reach daily lower limit
2	走强	Zou Qiang	Trending high	敢死队	Gan Si Dui	Suicide squad
3	十只	Shi Zhi	Ten stocks	准确率	Zhun Que Lv	Accuracy
4	涨	Zhang	Rise	日盘	Ri Pan	Open hours market
5	抢反弹	Qiang Fan Tan	Trade before revert	跌	Die	Drop
6	拉升	La Sheng	Push up	不超	Bu Chao	Less than
7	发稿	Fa Gao	Report	全网	Quan Wang	All over the internet
8	早盘	Zao Pan	Morning market	十档	Shi Dang	Level 10
9	面上	Mian Shang	On the surface	净流入	Jing Liu Ru	Net inflow
10	日复盘	Ri Fu Pan	Daily market review	送股	Song Gu	Bonus share
11	首日	Shou Ri	First day	高频	Gao Ping	High frequency
12	快讯	Kuai Xun	Breaking news	全线	Quan Xian	Everywhere
13	起复盘	Qi Fu Pan	Market review	最低价	Zui Di Jia	Lowest price
14	首个	Shou Ge	First	减持	Jian Chi	Selling stock
15	股票交易	Gu Piao Jiao Yi	Stock trading	汇总	Hui Zong	Summary
16	预增	Yu Zeng	Rise before earning report	跌幅	Die Fu	Decline
17	举牌	Ju Pai	Initial Public Offering	弱	Ruo	Weak
18	上证指数	Shang Zheng Zhi Shu	SSEC index	大跌	Da Die	Fall sharply
19	差额	Cha E	Difference	涉嫌	She Xian	Involved in
20	大阳线	Da Yang Xian	Rise intraday	终止	Zhong Zhi	Terminate

Note. These words are selected as a group to best augment the prediction by latent factors.

The words are ranked by their sentiment level. The top five sentiment-charged words for positive returns are

> FarmPredict: 涨停 (reached daily upper limit), 走强 (trending high), 十只 (10 stocks), 涨 (rise), 抢反弹 (trade before a rebound);

and for negative returns, they are

> FarmPredict: 跌停 (drop to the lower limit), 敢死队 (suicide squad), 准确率 (accuracy), 日盘 (open hours), 跌 (drop).

Results in Figure 4 and Table 3 indicate that unlike previous studies that only cover trading-related information, FarmPredict would capture all information of the article to select coordinated words, resulting more regularly in "nonsentiment" words, such as "十只 (10 stocks)" and "敢死队 (suicide squad)." Because there is particular language and writing mannerisms of each human being, not only general sentiment-charged words but also fixed collations and metaphors may be used to express and state comments and opinions in news. For instance, we barely find the word "敢死 队 (suicide squad)" in any sentiment dictionary from previous studies, but when writing articles, the reporter and editor usually analogize the monetary inflow in a depressed stock market to "suicide squad." Hence, we found that it has a strong predictive power of negative returns.

Another interesting finding is that top positive sentiment-charged words in Chinese stock markets are more "trading related," whereas previous literature about the U.S. market concluded a more "value-related" result (Ke et al. 2019). It also matches the current condition in Chinese stock markets that individual investors play a more critical role in market trading and are more likely to be influenced by trading-related news. Therefore, instead of a value-related signal, positive trading-related news of stocks will be more effective in explaining asset price changes in Chinese stock markets, known as the "herding effect." This result also demonstrates a relatively lower efficiency in Chinese stock markets. Unlike positive words, there is a more "value-related" phenomenon in the negative part, with more legal-related words, such as "involved in" and "fraud," which are traditional influencing factors on asset pricing. This result illustrates more rational behavior and implies greater similarity to the U.S. market.[20] Because a short sale is constrained and costly in China, mostly conducted by institutional investors or professionals, asset pricing information provides a stronger signal for driving market trading

4.1.2. Do Sentiments Predict Returns? Even though we have tested the consistency of our sentiment-charged words and the sentiments, it is still critical to directly validate whether our calculated sentiment scores have any prediction power on the returns. Based on our training target, we would expect that our sentiment scores can predict the beta-adjusted returns of their associated stocks. However, this process should not capture the information of the whole market, thus resulting in much weaker prediction power on market returns.

We first conducted the regression by forming panel data for the beta-adjusted returns of stocks from January 2015 to December 2019. The multiple regression is the following, in which we suppress the regression coefficients:[21]

$$Return_{it} = Sentiment_{i,t-1} + Return_{i,t-s} + X_{it} + \mu_t + \epsilon_{it},$$

where $Return_{it}$ is the beta-adjusted return of stock i in day t; $Sentiment_{i,t-1}$ is the corresponding sentiment score of stock i in day $t-1$;[22] X_{it} covers other stock-level variables to be specified; and μ_t is time (day) fixed effects capturing the time-related daily effect, such as market conditions and economic growth. As our sentiment score is trained with the stock-related news, there is a possible endogeneity issue that the news we use is driven by the beta-adjusted returns (i.e., the news might be reported after the extremely high/low beta-adjusted return occurred). The use of lagged returns mitigates this endogeneity issue between returns and the sentiment scores. We also controlled other stock-level variables, including stock size, price to book, return beta and alpha of the last year, stock volatility, and earnings surprises, to achieve a robust estimation. Moreover, because the beta-adjusted returns might be correlated with their past data, we added the one-week lagged returns as the control variables.

We gradually added the control variables into the model to test the robustness of the correlation. As shown in Table 4, there is a significant positive correlation between beta-adjusted return and the sentiment score. This positive correlation stays robustly significant only with the coefficient turning smaller after controlling the lagged returns and other stock-level variables. It can be seen from Table 4 that our sentiment scores are highly correlated with the beta-adjusted returns of each corresponding stock with a strong multiple R^2 and thus, can be applied to build portfolios with high beta-adjusted returns.

Even though Table 4 provides strong evidence of the prediction power of our sentiment scores on their associated stock returns, it is still essential to check if we captured the genuine specific features of stocks on that day but not the global attributes and information of the whole market. With this goal, we then conducted a similar regression analysis between daily market returns, daily average sentiment scores, and their dispersions, which are calculated based on all sentiment scores for the articles published on that day. We took daily returns of market indices in Shanghai

Table 4. Correlation Between Sentiment Score and Stock Beta-Adjusted Return

	FarmPredict			
Beta-adjusted return	(1)	(2)	(3)	(4)
$sentiment_{i,t-1}$	0.350***	0.208***	0.193***	0.193***
	(0.011)	(0.036)	(0.045)	(0.045)
Lagged returns		Yes	Yes	Yes
Control variables			Yes	Yes
Earnings surprises				Yes
Time fixed effect	Yes	Yes	Yes	Yes
Adjusted R^2	0.007	0.023	0.031	0.031

Notes. This table presents the estimation results of equation $Return_{it} = Sentiment_{i,t-1} + Return_{i,t-s} + X_{it} + \mu_t + \epsilon_{it}$. The outcome variable is the beta-adjusted return of stock i in day t. All standard errors are clustered by stocks. The scores are normalized and centered at 50. Statistical significance is indicated with asterisks.
***$p < 0.01$.

and Shenzhen stock markets to form time-series data from January 2015 to December 2019 and fit the following regression model:

$$Return_t = AveSentiment_{t-1} + \mathrm{DISP}_{t-1} + Return_{t-s} + Return_{t-l} + X_t + D_{year} + D_{month} + \epsilon_t,$$

where $Return_t$ is the return of the Commodity Selection Index (CSI) 300 index (Shanghai and Shenzhen 300 index); $AveSentiment_{t-1}$ is the daily average sentiment score; DISP_{t-1} is the dispersion variable of the score to control the variation represented by the standard deviation;[23] and D_{year} and D_{month} are year and month fixed effects to control yearly and monthly related trends, respectively. To mitigate the endogeneity issue between sentiment and market return, we also used the lagged terms of sentiment. We controlled the short-horizon lagged $Return_{t-s}$, long-horizon accumulated terms $Return_{t-l}$ of market returns, and other market variables X_t, including valuation measures and levels of interest rates in our models, to provide a robust estimation.

The results are shown in Table 5. We studied the correlation between the sentiment scores and market returns by sequentially adding the lagged terms. The results in Table 5 reveal that unlike Table 4, none of the results could provide evidence of the predictability of sentiment scores on the market returns. These nonsignificant results in Table 5 meet our expectation; because the sentiment scores are trained based on the beta-adjusted returns of individual stocks, a well-tuned model will only capture information about the individual stock but not the market. Both of the results in Tables 4 and 5 validate the performance of our model in extracting stock-level information from the news and neglect the global information of the market.

4.1.3. Event Study on Sentiment Scores.

In this subsection, we conducted an event study to see whether there is a significant reaction of individual stocks to sentiment scores. We treated the occurrence of sentiment score as an "event" and took a subsample that covered 14 days before and after the occurring date. Therefore, we can observe the pattern of beta-adjusted return change caused by the news and stock sentiment in this panel data. Then, we conducted the regression as follows:

$$Return_{it} = \sum_{p=-13}^{14} \beta_p Day_{ip} + \delta_i + \mu_t + \epsilon_{it}, \tag{4.1}$$

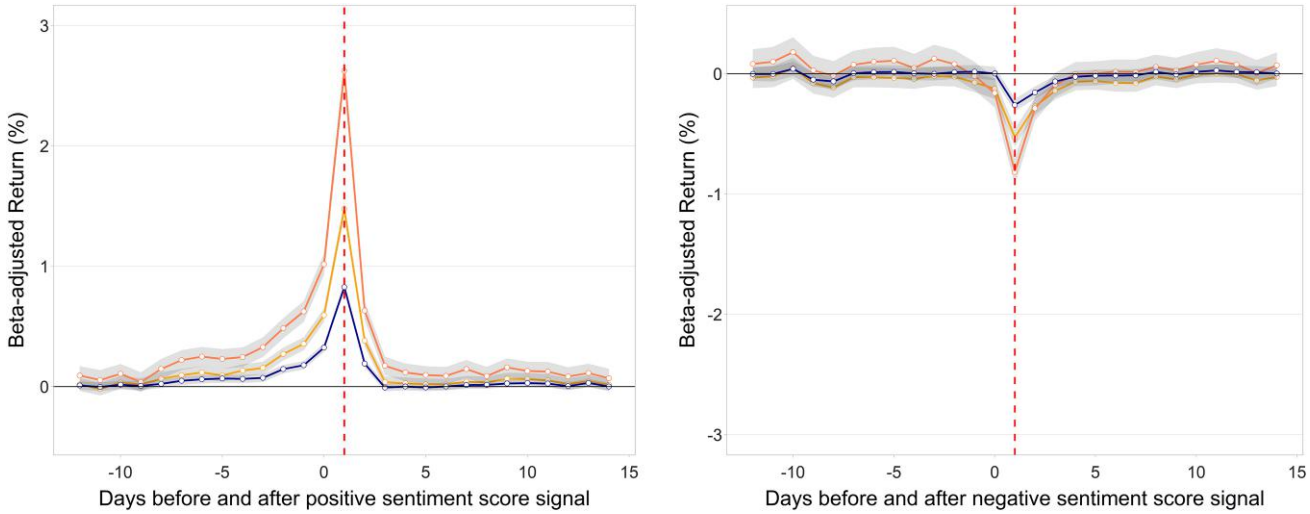
where $Return_{it}$ is the beta-adjusted return of stock i in day t; Day_{ip} are indicators of days before and after the sentiment occurs, of which the range is -13 to 14;[24] and δ_i and μ_t are stock individual and day fixed effects to control the heterogeneity in stock and date, respectively.

Table 5. Correlation Between Sentiment Score and Market Return

Return	(1)	(2)	(3)	(4)	(5)	(6)
$AveSentiment_{t-1}$	0.006	0.053	0.004	−0.004	−0.031	−0.045
	(0.118)	(0.125)	(0.129)	(0.131)	(0.134)	(0.144)
$DISP_{t-1}$		−0.251	−0.286	−0.310	−0.187	−0.132
		(0.226)	(0.228)	(0.228)	(0.238)	(0.261)
Market variables			Yes	Yes	Yes	Yes
Short horizon lagged return				Yes	Yes	Yes
Long horizon accumulated return					Yes	Yes
Month fixed effect						Yes
Year fixed effect	Yes	Yes	Yes	Yes	Yes	Yes
Adjusted R^2	−0.001	−0.001	0.002	0.008	0.016	0.011

Notes. This table presents the correlation between market return and the mean sentiment score. The outcome variable is the market return of the CSI 300 index on day t. The dispersion variable is represented by the standard deviation of the daily sentiment scores. We also controlled short-term lagged returns for five days and long-horizon accumulated returns for three months, six months, and one year. The market variables, including Cyclically Adjusted Price Earnings and interest rate, are also added.

Figure 5. (Color online) Event Study on Beta-Adjusted Return Before and After the News Announcement



Notes. The horizontal axis represents the days before and after news announcements, and the vertical axis is the beta-adjusted return during that day. We set day 1 as the day of the event (news) occurring. The coral, orange, and blue lines represent subsamples of the top 25%, 50%, and 100% positive/negative news. The white circles are the point estimates of the mean beta-adjusted return (estimated β_p in model (4.1)), and the bands around the circles indicate the 95% confidence intervals of the point estimates. This figure illustrates the trend of beta-adjusted returns before and after news announcements.

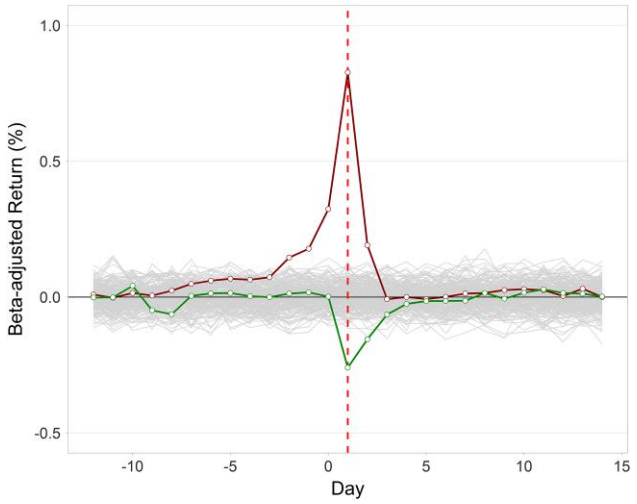
Model (4.1) provides straightforward results on how the markets and stocks anticipate (before) and react (after) to the sentiment scores. Figure 5 depicts the results of fitting model (4.1) separately on the top 25%, 50%, and full-sample positive/negative news. The results align well with each other and show significant heterogeneous mechanisms between positive and negative news. For the positive sentiments, the beta-adjusted returns start to increase and reach a relatively high level about seven days before the sentiment occurs. Such a trend is stronger for more positive news as the top positive news will lead to a higher return in Figure 5. The highest impact is on the day that news arrives, with an average of 83 bps for the full sample. Consistent with the discussion on the words we extracted from the news, positive news in the Chinese stock markets mainly covers trading-related reports. Another possible reason is that the information is leaked to market participants, leading to an increase in return before the news occurs.

However, for negative news, we did not observe this phenomenon that returns decrease prior to a news announcement. The beta-adjusted return is only negative when news occurred, which has an average impact of 26 bps for the full sample. This aligns well with the short-sale restrictions in the Chinese markets. Even if negative news is leaked or anticipated, transactions are hard to take place. It is also consistent with the result in Figure 5 that positive news has a bigger impact on stock returns than negative news, contrary to the behavior in the U.S. equity market.

For the beta-adjusted returns after the news announcement, we found similar patterns for both positive and negative news, with existence periods of two and three days, respectively. Beta-adjusted returns after this period are statistically insignificant from zero for both groups. The results show an arbitrage opportunity for portfolios built on the day after news announcements. Therefore, the results of this event study also provide a mechanism for why our constructed portfolios in the next subsection can achieve high beta-adjusted returns based on the sentiment scores.

4.1.4. Placebo Test. In this subsection, we conducted a placebo test for our event study to test if this specific trend of beta-adjusted returns is caused by the event as measured by the sentiment scores in this paper. To evaluate this, we randomly pick a subsample with continuous 28 days from each stock, the same length as that in the previous event study, from the data not overlapping with the event period. Then, we reran the event study regression on this new random sample and replicated it 200 times to see if the significantly outperformed returns will occur. This results in 200 curves, depicted in Figure 6. The gray area is the accumulated estimation results of each replication, showing a distribution with a mean of zero. The results in Figure 5 are superimposed in Figure 6 for comparisons. This result boosts our confidence that the results in Figure 5 are robust and genuine and specifically caused by the news and reports. Moreover, we can observe that beta-adjusted returns after the initial day

Figure 6. (Color online) Placebo Test of Sentiment Score



Notes. The light gray lines are the point estimates based on 200 experiments from fitting the model $Return_{it} = \sum_{p=-13}^{14} \beta_p Day_p + \delta_i + \mu_t + \epsilon_{it}$ on the subsample by removing observations of stocks affected by the news (sentiment). All others are the same as those in Figure 5.

of new announcements still stand out from the placebo returns, providing a tradable portfolio-building strategy, which will be introduced in the next section.

4.2. Portfolio Performance

We also tested the models by building stock portfolios based on their predicted scores. Portfolios are built and tested in each rolling window as follows. The model would gather all the articles from the previous market close (3:00 p.m. on day $t-1$) to the current market close (3:00 p.m. on day t) and calculate the corresponding score of each article-related stock; hence, our strategy would cover all news that occurred 24 hours before the market close on day t. Then, we invest by longing 50 stocks with the highest scores and shorting 50 stocks with the lowest scores.[25] Suppose that there are fewer than 50 stocks with positive/negative signals. In that case, the unallocated capital will be kept as cash (with no interest), so the portfolio's total capital exposure would never be greater than 100%. We form our position at the closing auction and close it at the second trading day's closing auction (day $t+1$). Under an EW set, we long and short each stock with the same fixed 1% total capital exposure each day.

We also tested the portfolio performance under a VW set, where the weights are set to be proportional to stocks' total market capitalization on the prior day. Such a portfolio would put larger weights on large-cap stocks compared with small-cap ones. Usually, there are more informed investors trading large-cap stocks, leading to more efficient prices, better liquidity in trading, and fewer returns (Ke et al. 2019). We anticipated that it is less affected by new sentiments.

4.2.1. Transaction Fee and Price Limit.

There are significant transaction costs and taxes charged by the exchanges or stock retailers for the daily portfolio strategies in China. Transaction costs of trading in Chinese stock markets are made up of the following three main components.

1. Stamp duty is 0.1% of the total capital transaction amount. Only sellers are charged. This is equivalent to 10 bps costs in our portfolio if all positions are liquidated the next day.

2. Transfer fee is one Chinese Yuan (CNY) for each 1,000 shares traded and is charged to both buyers and sellers; it is only charged on stocks traded on the Shanghai Stock Exchange. Thus, there is a 1 bps combined cost (buy and then sell on the Shanghai Stock Exchange) for a stock with a price of 20 CNY per share.

3. Trade commission ranges from 0.01% to 0.02% for each transaction and is charged by stock retailers on both sides of a trade. Typical rates are around 1 bps.

In a typical case with a stock price above 20 CNY (most stocks are above this price), each trade we made (buy and sell combined) incurs 10 bps in stamp duty, about 1 bps in transfer fees (even the number of stocks in the Shanghai Market is about 35% of the whole), and 2 bps in trade commission. So, only trades with a positive expected return of over 13 bps daily are profitable under these conditions.

Another issue is that China imposes a 10% price limit in its equity market,[26] serving as a market stabilization tool. On each trading day, no order can be placed or traded at prices outside the $\pm 10\%$ range of its previous closing price. Moreover, once the price of one stock reaches the price limit, it becomes barely tradable, and only a fraction of stock orders might eventually be executed because all trades happen at the same limit prices. Hence, this restriction would affect our strategy by making stocks at limits difficult to trade.[27]

As it would require high-frequency order data to realize how many orders could be executed for a stock at the price limit, we are unable to provide a precise result of the realistic portfolio performance. To ameliorate this issue, we provided its upper and lower bounds, which correspond to the extreme situations where all or none of the stocks at the price limit were traded. It is worth noting that our portfolio strategy could be improved by using higher-frequency data (i.e. long/short right after the news announcements) hold the capital when no stocks are eligible for investment to avoid the high trading fee, etc. Nevertheless, the portfolio performance with the current evaluation strategy would still reveal how much our model learned from the text data and provide a fair way for model comparison (see Section 5.2).

4.2.2. Basic Performance.

The portfolio returns for the EW strategy are computed based on \$100 invested each day: investing \$1 on each of long or short positions and

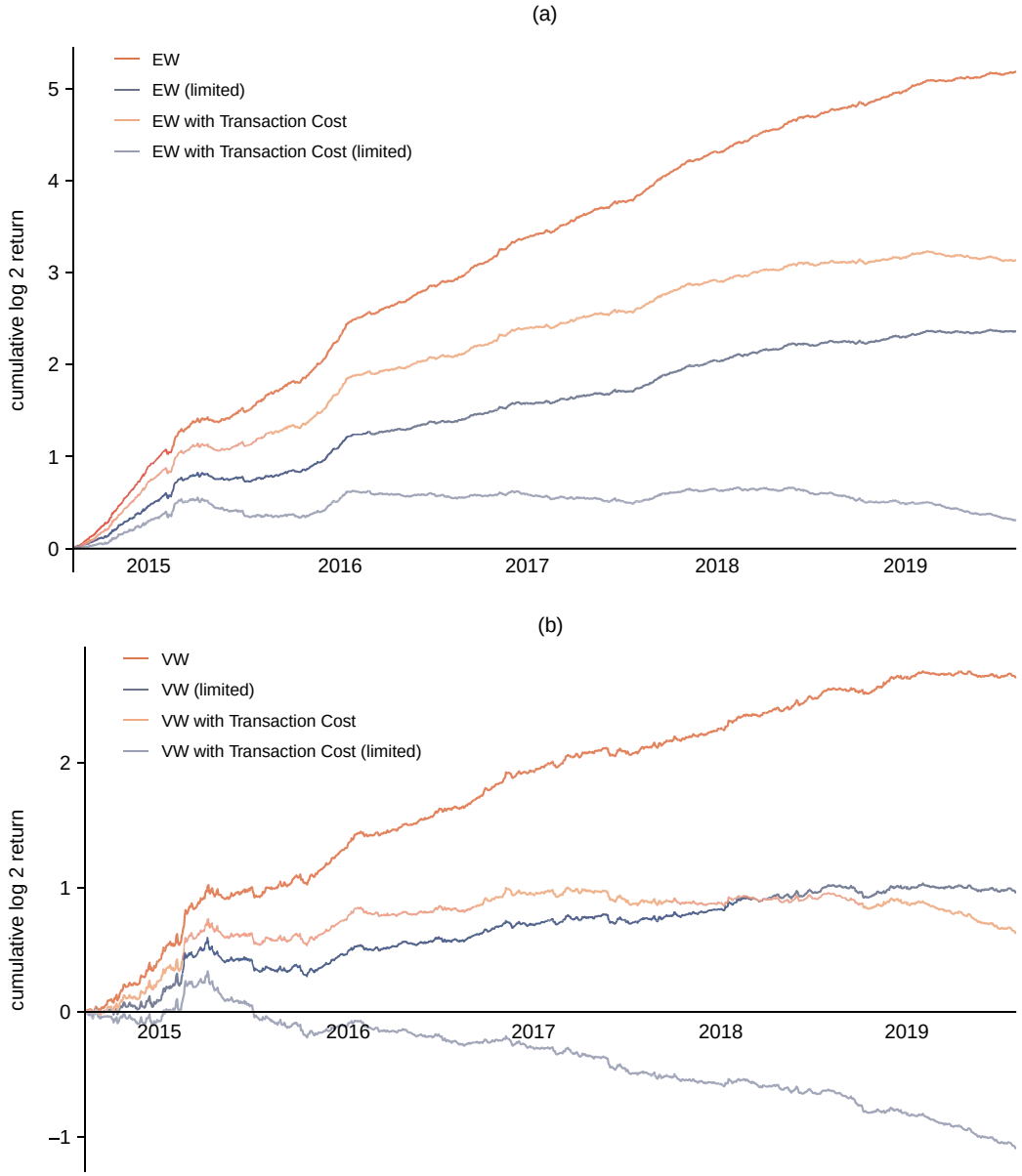
adding up the total daily gains divided by 100. Because this is a long-short portfolio, the actual capital expenditure is much lower than 100, yielding even better performances. A similar computation of portfolio returns is applied to the VW strategy.

To accurately account for the transaction fee, we first calculated the daily average of the total changed proportion of portfolios in Equation (4.2) as the average turnover ratio. The total portfolio weight $\mathbf{w}_t$ is no greater than one by construction (typically equal to one), and the case $\|\mathbf{w}_{t+1} - \mathbf{w}_t\|_1 = 2$ implies that the portfolios are totally different between day t and day $t+1$. The portfolios between every two adjacent days are compared, and only their differences are traded. For simplicity, we ignored the changes of weights from t to $t+1$ because of stock price changes in turnover calculations:

$$\text{Average Turnover Ratio} := \frac{1}{2(T-1)} \sum_{t=1}^{T-1} \|\mathbf{w}_t - \mathbf{w}_{t+1}\|_1. \tag{4.2}$$

Figure 7 illustrates the basic performance of our model, where we compared cumulative log 2 returns with and without the transaction costs and price limit constraints. The detailed performance of the combined long-short,

Figure 7. (Color online) Cumulative log2 Returns of EW and VW Portfolios



Notes. Transaction costs and price limits are both considered. We assumed the extreme case where liquidity becomes strictly zero for stock-triggered price limits. (a) Cumulative log2 returns of the EW portfolio. (b) Cumulative log2 returns of the VW portfolio.

Table 6. Portfolio Performances from 2015 to 2019 of FarmPredict

	Upper bound (without price limits)				Lower bound (with price limits)			
	No Transac		With Transac		No Transac		With Transac	
Portfolio	SR	APR, %	SR	APR, %	SR	APR, %	SR	APR, %
EW								
L + S	8.51	105.16	5.32	54.35	4.26	38.70	0.21	4.30
L	3.86	75.76	4.36	52.44	1.74	20.18	0.17	4.21
S	1.14	16.30	−0.18	0.84	1.41	15.00	−0.31	−0.28
VW								
L + S	2.94	45.18	0.46	9.18	0.86	14.24	−1.21	−14.11
L	4.1	46.00	2.27	26.62	1.37	15.68	−0.23	0.31
S	−0.30	−0.60	−1.57	−13.81	−0.37	−1.28	−1.64	−14.40

Notes. The transaction cost is placed daily when components of the portfolio are changed. Transaction cost includes stamp duty, transfer fee, and trade commission in China. We assumed a 13 bps transaction cost for each buy and sell trade combined for the "With Transac" column. The turnover ratio is considered when calculating the returns. The upper and lower bounds indicate that all/no stocks at price limits are traded. L, long; L + S, long-short; S, short; Transac, transaction.

long-leg, and short-leg of the portfolio is shown in Table 6.

Table 6 demonstrated that the transaction fee would strongly affect the performance of FarmPredict, and the APR would reach 105% under a perfect condition (no transaction fee and price limits), but only 54% might be realized. Moreover, the price limits also lead to a significant decrease in portfolio performance: that the APR would drop from the ideal setting (all orders are executed) of 54.35% to the worse scenario (no orders are executed) of 4.30%. Such a result would indicate that there are stock price-related signals residing in Chinese news texts as our model would capture the most positive (trigger the price limits) stocks at a daily level.[28] The Sharpe ratio changes with the returns from 5.32 to 0.21 under the EW set. According to Chen et al. (2015), the relative rational SR is about 2.8, which allocates in our range. Another observation is that the portfolio returns are mostly realized from the long leg rather than the short leg, which would even be negative under the VW set. Such a finding is also in line with those presented in Figure 5.

Despite the 45.18% APR performance of value-weighted portfolios, once transaction costs and price limits are involved, the strategy is no longer profitable. This suggests that large-cap stocks are more popular and better studied, so their prices are less affected by the arrival of financial news; hence, the high trading fee erases the profits of the news when trading daily. Such a result for the Chinese market is consistent with those in the U.S. market obtained by Ke et al. (2019).

4.2.3. Return Compositions and Market Risks.

To better understand the allocation strategy, we studied the components of its returns and risks. We introduced measures to decompose and evaluate a portfolio's idiosyncratic return and pricing factor exposure and used them to analyze FarmPredict's returns and risks from its long leg, short leg, and market movements.

Stock short-term movements induced by market conditions are usually thought of as orthogonal to the stock's fundamentals or stock-specific signals. We used the four-factor model in Carhart (1997) to evaluate alphas rather than raw returns. Here, beta represents the stock's exposure to market movements. Estimating alphas helps us to understand whether the return associated with the news sentiment strategy is driven by exposure to common risk factors. We used the linear regression in Carhart (1997) as follows:

$$R_{p,t} - R_f = \alpha + \beta_1 \mathrm{MKT}_t + \beta_2 \mathrm{SMB}_t + \beta_3 \mathrm{HML}_t + \beta_4 \mathrm{MOM}_t + \varepsilon_{it},$$

where the R_f is the risk-free rate; MKT_t, SMB_t, HML_t are the three factors introduced in Fama and French (1993) covering the market, small minus big, and high minus low factors; and MOM_t is the momentum factor that is estimated by the difference between the return rates of the most and least profitable stocks during the past 11 months. The alphas can then be estimated.

To further quantify the relationship of our returns to the market, we propose the following R^2 measure to account for the amount of variance in portfolio returns that are related to the market. Based on the decomposition of returns R_p, we define R^2_{factor}[29] as the proportion of variance in returns from the market as

$$R^2_{factor} = \frac{\sum_t (\beta_1 \mathrm{MKT}_t + \beta_2 \mathrm{SMB}_t + \beta_3 \mathrm{HML}_t + \beta_4 \mathrm{MOM}_t + R_f)^2}{\sum_t [R_{p,t} - \mathrm{ave}(R_{p,t})]^2}.$$

Results on the portfolios based on our FarmPredict model are shown in Table 7. All transaction costs are considered. Only 6.92% of the overall variance is related

Table 7. Characteristics of EW and VW Portfolios Based on FarmPredict

Portfolio	SR	APR	Alpha APR	R^2_{factor}	Daily return, bps
Upper bound					
EW					
L + S	5.32	54.35	51.06	6.92	17.7
L	4.36	52.44	47.08	34.25	17.2
S	−0.18	0.84	2.72	27.18	0.3
VW					
L + S	0.46	9.18	7.06	5.48	3.6
L	2.27	26.62	24.94	3.32	9.6
S	−1.57	−13.81	−14.32	7.28	−6.1
Lower bound					
EW					
L + S	0.21	4.30	2.01	5.15	1.7
L	0.17	4.21	0.51	35.95	1.7
S	−0.31	−0.28	1.50	25.68	−0.
VW					
L + S	−1.21	−14.11	−15.84	5.08	−6.2
L	−0.23	0.31	−1.06	3.53	−0.1
S	−1.64	−14.40	−14.94	8.64	−6.3

Notes. The testing period ranges from 2015 to 2019. Sharpe ratios, daily and annualized average returns, robust alpha, and market exposures are reported. Separated returns of the short and long legs of both portfolios are reported as well. L, long; L + S, long-short; S, short.

to the market because market exposures from longs and shorts cancel out when combined. The long and short legs themselves, as expected, assume large market exposures from 27.18% to 34.25%.

The performances of portfolios in both regular returns and factor-adjusted returns[30] from 2015 to 2019 are shown in Figure 8. The curves of cumulative raw returns and factor-adjusted returns (alphas) are very close in the figure, indicating that the portfolio is minimally exposed to market risks. The long and short legs cancel out each other's short-term variations and contributed to the overall portfolios in different periods.

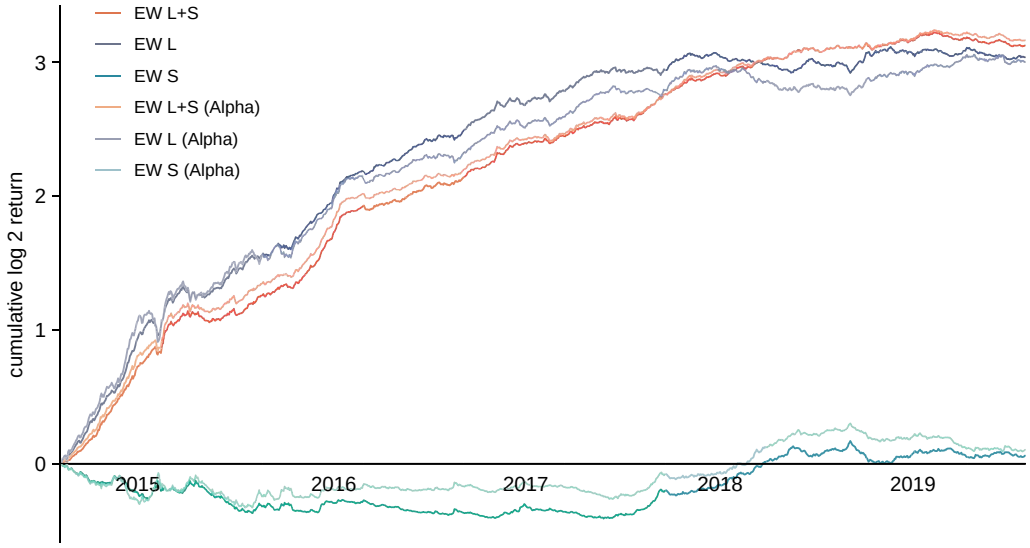
5. Model Discussion and Comparison

Besides the performance analysis of both return prediction and portfolio building, we still would like to assess empirically the methodological novelty of FarmPredict, say by augmenting the prediction model with factors and covering advantages from both word selection (residuals) and word clustering (factor). To do that, we first discussed FarmPredict by isolating the "contribution" of factors and idiosyncratic residuals for prediction and then compared it with other textual models.

5.1. Discussion of the Model

5.1.1. Factors vs. Residuals. As we mentioned in Section 2, the most novel part of FarmPredict is to (without supervision) convert high-dimensional variables into factors and idiosyncratic residuals. Unlike previous factor models, FarmPredict takes the idiosyncratic residuals into models instead of only using factors, which avoids the information loss by principal components in the dimensionality reduction and results in significant improvements in prediction. Nevertheless, we are still curious about the "contribution" of each part in our case. Hence, we conducted Equation (2.6) using factors and idiosyncratic residuals separately. Such a process

Figure 8. (Color online) Cumulative log2 Returns of Long-Short, Long-Only, and Short-Only Strategy and Their Associated Factor-Adjusted Returns from 2015 to 2019



Notes. Our portfolio has little correlation with the market, with curves of the beta-adjusted return almost perfectly overlapping with the raw ones. Investing on both the long and short sides greatly helped smooth out market volatility. L, long; L + S, long-short; S, short.

Table 8. Comparison of Different Components in FarmPredict

Components	R^2, %	Daily return, bps	Difference in return, bps
Full model	4.21	17.8	—
Factors only	−0.12	−8.2	26***
Residuals only	4.21	16.4	1

Notes. This table shows the fitting result and the portfolio performance using different components of FarmPredict. We mainly compare the R^2 calculated by the test sample's combined result, the daily return with transaction fees, and the difference in daily return. Statistical significance is indicated with asterisks.
***$p < 0.01$.

would allow us to further detect the importance of each part and be insightful for model building in textual analysis. Note that idiosyncratic residuals use both data from the original data X and latent factors f. Even though the factor part does not directly contribute to the prediction, it contributes to the prediction through the idiosyncratic residuals, and the model is still different from using LASSO on X.

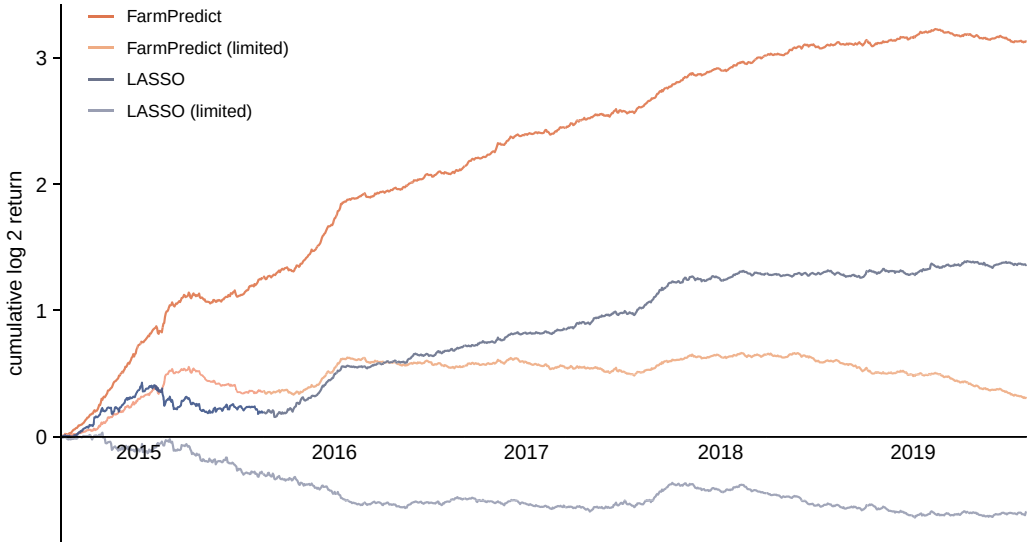
We set the EW with transaction costs but no price limits as the baseline for model comparison. Table 8 reports the prediction results of the full model, the factor-only model, and the residual-only model. We focused on performances of both the prediction and the portfolio building.[31] Table 8 provides solid support for FarmPredict in high-dimensional studies: that idiosyncratic residuals matter in our case, whereas covering factors would help improve the prediction. The model and portfolio performances are similar to the full and residual-only models, with a slight difference in portfolio performance.

However, Table 8 would lead to another concern: that FarmPredict might collapse into an LASSO model because residuals almost account for all the contributions. Therefore, we further compared FarmPredict with the naive LASSO using X with ℓ_1 to learn finance. Figure 9 compares portfolio performance between naive LASSO and FarmPredict.[32] Conditions with and without price limits are presented for a robust comparison. Figure 9 proves that in terms of portfolio building, FarmPredict did not collapse into a naive LASSO model and that factor augmentation would significantly benefit the prediction, with an R^2 increase slightly from 3.99% (LASSO) to 4.21% (FarmPredict) and a 37.4% improvement in APR.

Results in Figure 9 and Table 8 are very insightful for textual analysis; instead of only doing dimension reduction, which is the main focus of most factor-based studies, or word selection process (i.e., LASSO), covering both elements, such as using factors to augment the predictors, results in better performance. Such results would shed light on model building in textual analysis: that studies might focus on taking advantage of both dimension reduction and word selection models.[33]

It is worth noting that although the residual u plays a critical role in our case, as FarmPredict covered both factors and residuals, the unsupervised tuning process would lead the model to an optimal balance between elements. Results in Table 8 would not imply that factor models are less efficient compared with other word selection models but demonstrate the importance to cover both of them in the textual building.

Figure 9. (Color online) Comparison with the Naive LASSO Model



Notes. This figure presents the comparison of accumulated returns of FarmPredict and LASSO strategies. The same data with FarmPredict are used for comparison. We tuned the hyperparameter λ in Equation (2.6) with the same tuning process of FarmPredict.

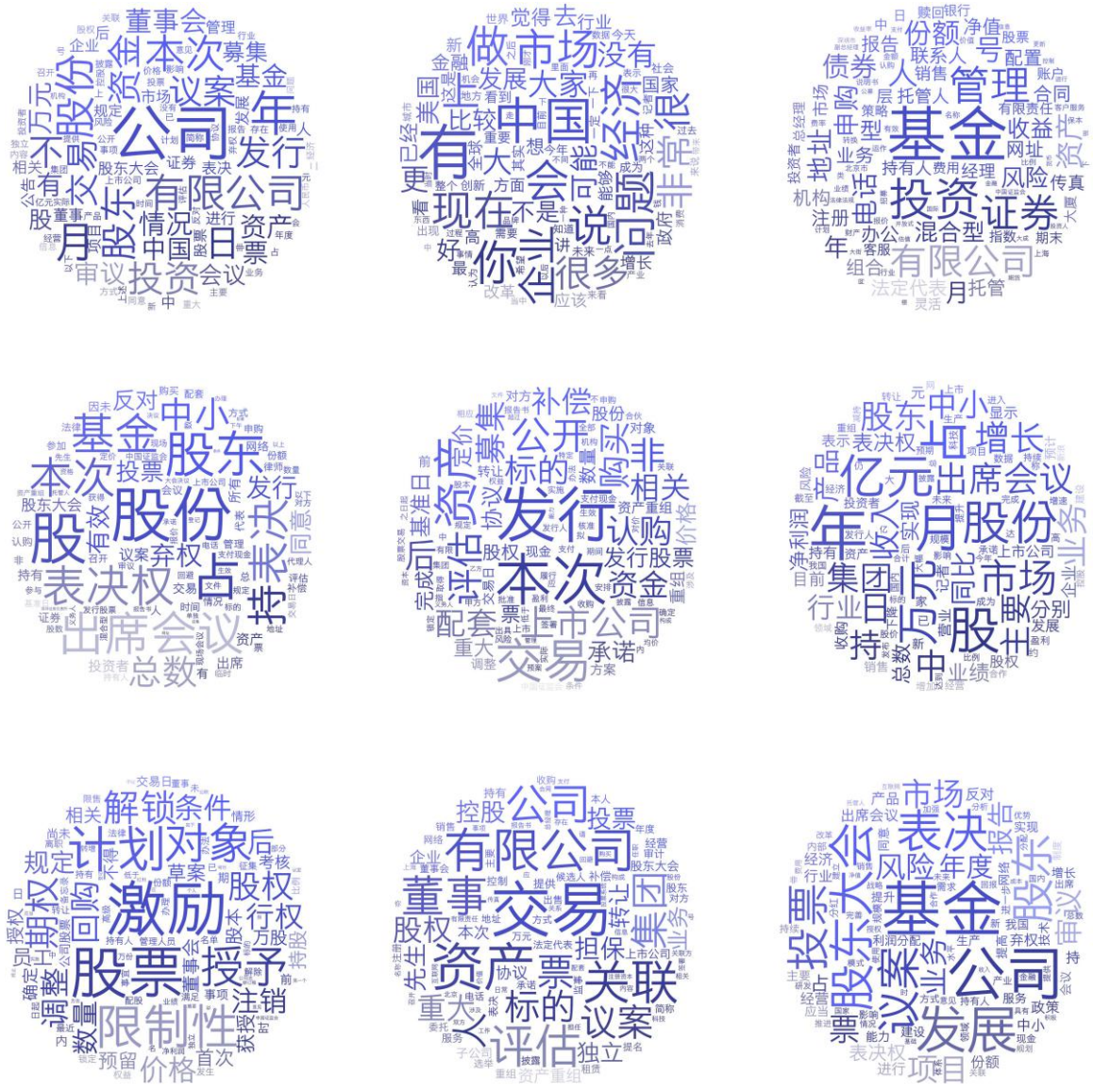
5.1.2. Content of Factors. We further studied the content of factors using the loading matrix mentioned in Equation (2.2) for a better understanding of the information that FarmPredict captured. Figure 10 presents the content of nine factors after removing the function words.[34] We provide the translation of the words in each figure in the online appendix. These factors can be further labeled into topics based on the top-weighted words in each factor, including firm, Chinese economy, funds, cooperate governance, Initial Public Offering, earnings, incentive, restructuring, and others. The labeling is not the main target of FarmPredict but simply provides some economic understanding of each topic. The word clouds in Figure 10 also demonstrate that our model could decompose the Chinese text into easily interpretable groups with specific topics, such as the economy and policy in China. Similar results are also found in Larsen and Thorsrud (2017).

Because most of these topics are neutral, simply extracting these topics out would not help predict returns. This result also shows a potential mechanism of FarmPredict; instead of relying on tons of strong topics or a selection of words, FarmPredict focused on the sentiment-charged residuals with (weak) topics.

5.2. Model Comparison

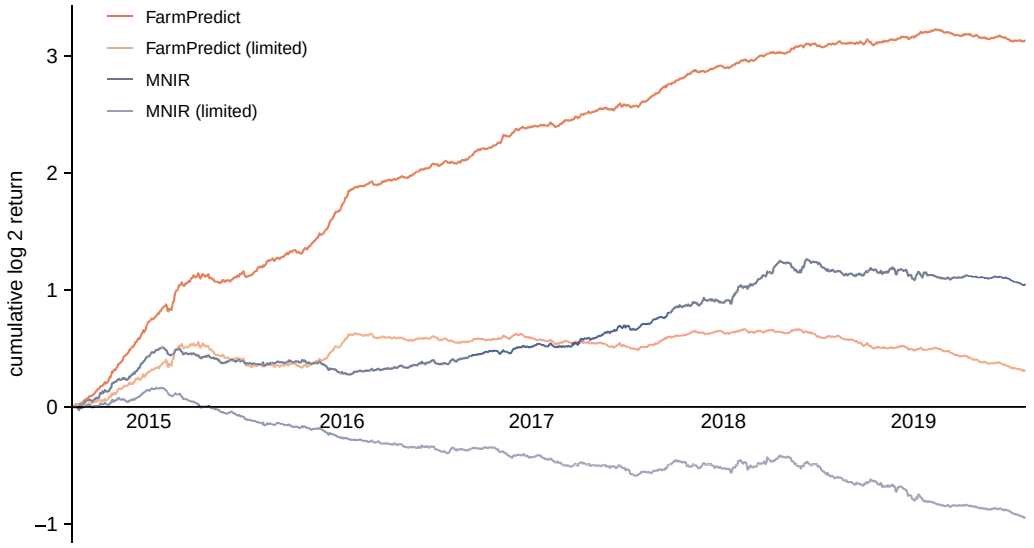
5.2.1. Word Selection Model. Despite that the discussion section has clearly shown how factor augmentation would benefit prediction, to demonstrate the advantage of FarmPredict covering both factor and individual

Figure 10. (Color online) Content of Factors



Note. This figure presents the word clouds of nine factors without the function words using the elements in the loading matrix **B**.

Figure 11. (Color online) Comparison with MNIR



Notes. This figure presents the comparison of accumulated returns of FarmPredict and MNIR introduced in Taddy (2013). The hyperparameters of MNIR are tuned with the same tuning process as FarmPredict.

words, we further compared it with other statistical models by word selection and clustering.

5.2.1.1. MNIR. First, we implemented MNIR, which is proven as a useful tool in textual analysis, specifically to solve the high-dimensionality issue by term selecting (Taddy 2013). MNIR focused on the hidden sentiment (topic) of the text and transformed the text-sentiment relationship into a uniformed multinomial inverse regression problem. The "Gamma-Lasso" scheme in Taddy (2013) would yield a stable and effective approach to MNIR estimation. We applied the R package developed in Taddy (2013) for implementation.[35]

Figure 11 demonstrated the comparison between FarmPredict and MNIR in terms of portfolio performance. We provided both upper and lower bounds of the result, and all transaction fees have been accounted for. Figure 11 illustrated that FarmPredict significantly outperformed MNIR in our case, speaking to the necessity of using both factor and residuals for prediction.

5.2.1.2. SESTM. We also compared FarmPredict with the state-of-the-art topic model, SESTM, introduced by Ke et al. (2019). It assumes that each article is a mixture of two topics—positive and negative—and uses the mixture probability p_i to indicate the positive sentiment on the ith article, with one being the most positive and zero being the most negative. Naturally, p_i is expected to be positively associated with return Y_i.

Assume sentiment-neutral vocabulary $\mathbf{N}$ is independent of either score p_i or return Y_i given the sentiment-charged words $\mathbf{S}$. Let s_i be the number of sentiment-charged words in article i. It assumes that the word count $\mathbf{d}_{i,\mathbf{S}}$ follows a multinomial distribution, which shares the same statistical thought in Taddy (2013):

$$\mathbf{d}_{i,\mathbf{S}} \sim \text{Multinomial}(s_i, p_i\boldsymbol{\theta}_+ + (1-p_i)\boldsymbol{\theta}_-),$$

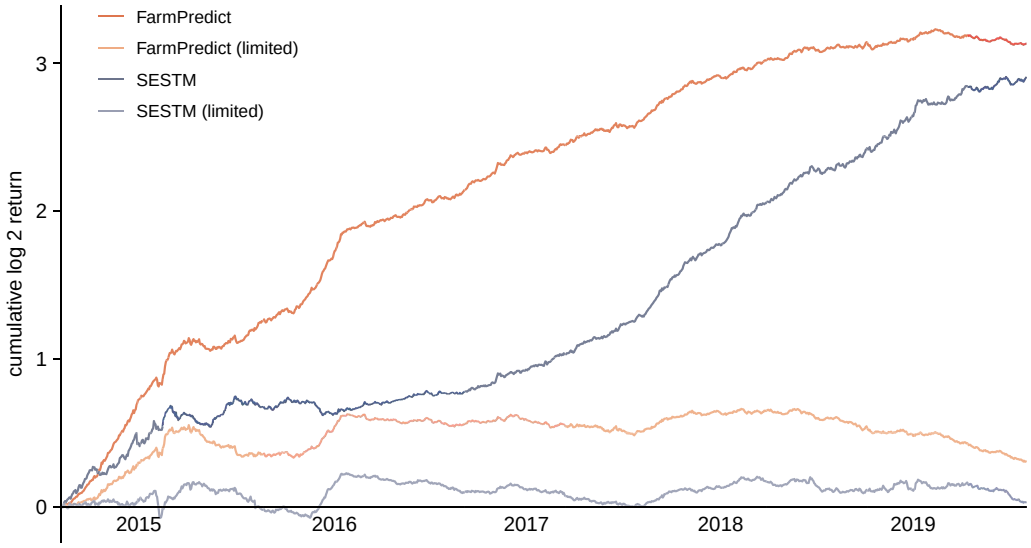
where $\boldsymbol{\theta}_+$ and $\boldsymbol{\theta}_-$ are two parameter vectors of dimension $|\mathbf{S}|$, indicating the probabilities of occurrences of sentiment-charged words $\mathbf{S}$ in a purely positive or negative article.

Learning sentiments from a set of training data $\{\mathbf{d}_i, Y_i\}_{i=1}^n$ consist of two main steps: learning the sentiment-charged vocabulary $\mathbf{S}$ and learning semantics of these words $\boldsymbol{\theta}_+$ and $\boldsymbol{\theta}_-$. The former uses the marginal screening techniques in Fan and Lv (2008), and the latter uses supervised learning with the assistance of the percentile ranking of the return Y_i in the training set. Once the sentiment-charged words and their semantics are learned, a new article's sentiment score p_i can be estimated using the maximum likelihood estimator.

We implemented the SESTM and tuned the hyperparameters using the same duration as FarmPredict. Details are shown in Section A.1 in the online appendix. Figure 12 provides the comparison results between FarmPredict and SESTM. The two models show quite a different trend in portfolio performance, with a slight difference in the final accumulated return. The outperformance of FarmPredict on SESTM would further enhance the advantage of using both factors and residuals for prediction.

5.2.2. Word Clustering Model. The previous result has demonstrated the advantage of FarmPredict compared with word selection models by augmenting predictors

Figure 12. (Color online) Comparison with SESTM

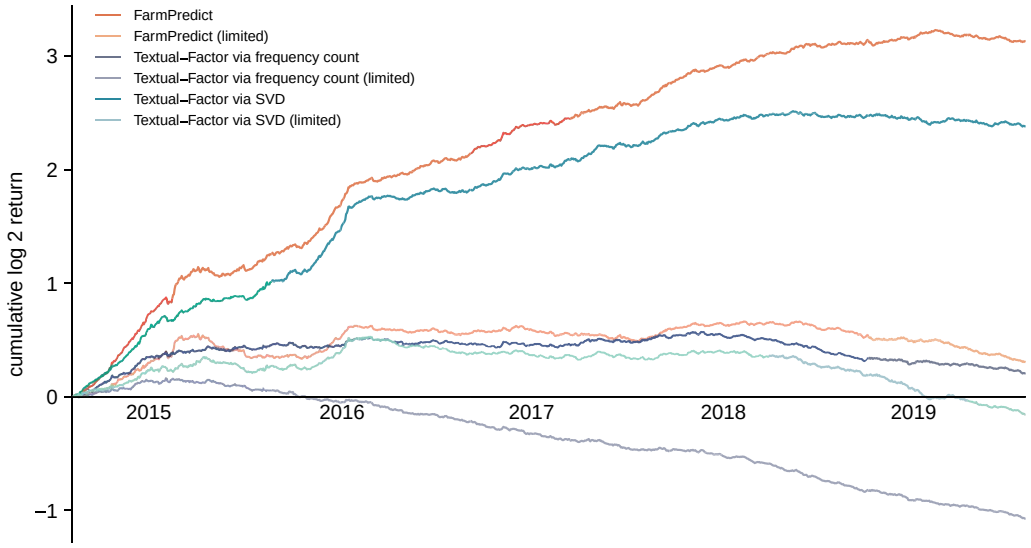


Note. This figure presents the comparison of accumulated returns of FarmPredict and SESTM.

with factors. Nevertheless, it would be essential to compare FarmPredict with word clustering methods. We applied the textual factor model in Cong et al. (2019) that reduces the word dimension via the word2-vec embedding approach. The model proposed by Cong et al. (2019) can be summarized in the following steps: (a) use a (pretrained) word2vec embedding model to transform the characters, words, and phrases within one document into vectors; (b) cluster the transformed data into factors (topics or groups) by a fast hierarchical algorithm; and (c) learn the topic factor f_i and importance of each factor x_i to reduce the dimension. Machine learning models can be applied afterward for prediction.

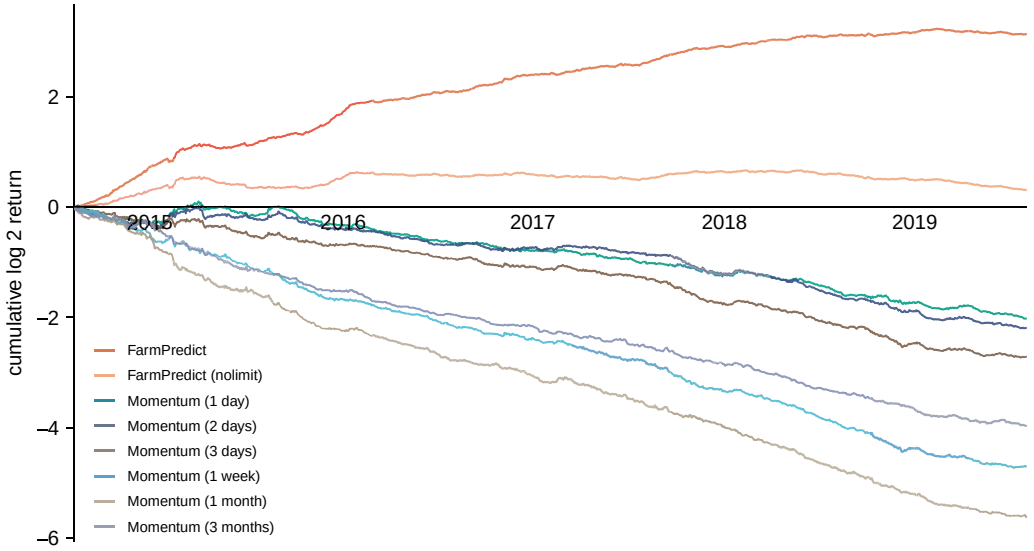
In Cong et al. (2019), pretrained word2vec models by Google are used. As we focused on Chinese text, we used the pretrained word2vec model trained by financial data to implement this model.[36] An LASSO approach is further conducted for prediction after the factorization process.[37] It is worth noting that although

Figure 13. (Color online) Comparison with the Word Clustering Model



Notes. This figure presents the comparison of accumulated returns of FarmPredict and the textual factor model in Cong et al. (2019). We used the codes on GitHub (https://github.com/textualfactor/Text_Analysis) for estimation. The hyperparameters of the textual factor model are tuned with the same tuning process of FarmPredict.

Figure 14. (Color online) Comparison with Momentum Strategies



Notes. This figure presents the comparison of accumulated returns of FarmPredict and traditional momentum strategies. We separately constructed portfolios based on the performance of each stock in the past one day, two days, three days, one week (lag = 5), one month (lag = 22), and three months (lag = 66) approximately considering the closing days of the market.

we both used the words "text factors," the methodology and content are very different. The factors (topics) in Cong et al. (2019) are formed by the clustering in step (b) using the pretrained word2vec model, and step (c) is just to help transform the data and reduce the dimension. Hence, the clustering or dimension-reduction process is highly dependent on the word2-vec approach, whereas FarmPredict relies on PCA, and factor contents rely on weights.

Figure 13 illustrates the model performance of the textual factor model in Cong et al. (2019) with a comparison of FarmPredict. We separately used the two algorithms (singular value decomposition (SVD) and frequency counts) in Cong et al. (2019) to load the data and calculate the importance of each factor. It shows that FarmPredict still outperforms the textual factor model, whereas the performance of the SVD algorithm is quite close to FarmPredict. This result further speaks to the advantage of FarmPredict for a double-robust estimation.

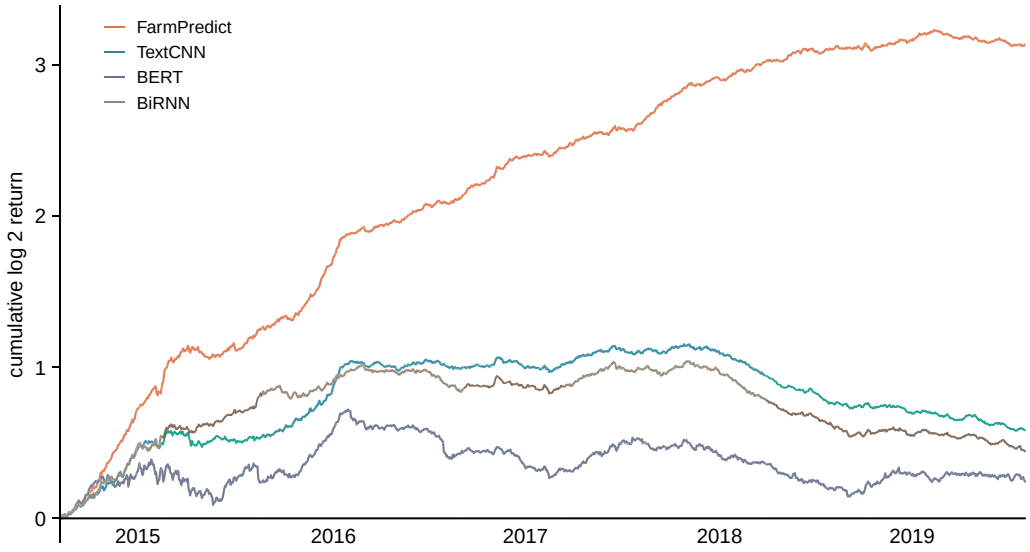
5.2.3. Traditional Strategy. As we have empirically compared the portfolio performance with other factor-based and word selection models, we still would like to include "traditional" models to emphasize the advantage of FarmPredict. Because Figure 5 illustrates a strong contemporaneous correlation in returns, we further compared our FarmPredict with momentum strategies, say building portfolios simply based on past returns.

We separately chose stocks based on the returns of the past 1 day, 2 days, 3 days, one week, one month, and three months, which cover approximately 1, 2, 3, 5, 22, and 66 trading days.[38] Then, we followed the EW rule introduced in Section 4.2 and built the stock portfolio based on the sorted momentums using past accumulated returns. The comparison result is shown in Figure 14. Figure 14 illustrates that the momentum strategies show a very weak performance in the Chinese stock market, where all portfolios with different duration result in negative (after accounting for the trading fee) accumulated returns. The outperformance against the momentum strategy demonstrates the advantage of FarmPredict as well as the relevance of the textual data.

5.2.4. Other Machine Learning Models. Despite the outperformance of FarmPredict compared with statistical models, we still would like to provide a more complete picture by introducing other state-of-the-art machine learning models. The recent success of these models in neutral language processing also suggests such points. Therefore, we implemented other models on the same data set, including BERT and neural network models.

For the BERT model, we fit and tuned the Chinese version of the pretrained BERT model with 12 layers of transformer encoder blocks with 768 hidden units and 12 self-attention heads. Then, we added a prediction layer after BERT separately using ordinary least squares (OLS), bidirectional recurrent neural networks (BiRNN), and text convolutional neural network (TextCNN).[39] A dropout layer with a probability of 0.5 is added before this prediction layer of all three models to prevent overfitting.

Figure 15. (Color online) Comparison with Other Machine Learning Models



Note. This figure presents the comparison of accumulated returns of FarmPredict and other machine learning models.

Because of the complexity of the BERT model with 12 layers, it would be sensitive to initialization, which easily converges to the local optimum. Therefore, we further used the pretrained model published by Google to initiate the parameters of the BERT model (Devlin et al. 2019). For neutral models, we implemented a BiRNN model that consists of one fixed embedding layer (from BERT pretrained or not), two bidirectional Long Short Term Memory networks layers with 100 hidden nodes in each layer, and one fully connected layer.

Unlike BiRNN, convolutional neural networks replace the fully connected layers in feed-forward neural networks with convolutional layers. TextCNN is built on a one-dimensional convolutional layer and max-over-time pooling. We built a TextCNN model with three convolutional layers with kernel sizes of three, four, and five separately. For model details, we first defined multiple one-dimensional convolution kernels to convolute the input. Then, TextCNN maximized the timing of all output channels and spliced the output values. Finally, one full connection layer would calculate the category output, which in our case, was the corresponding returns.

Figure 15 provided a straightforward comparison of portfolio performances of all models. Even though there are differences in the return level, the trends of BERT, BiRNN, and TextCNN are very close. Among all models, FarmPredict still shows the best performance in terms of portfolio-building strategy. A strong performance against these benchmark models further provides evidence of the practical relevance of our FarmPredict.[40]

5.3. Summary of the Model

As we compared FarmPredict with several traditional and state-of-the-art textual models, we would like to summarize the results to provide a clear view of how augmenting predictors using factors would improve prediction. Table 9 summarized the comparison between FarmPredict and other models in terms of fitting and portfolio (out-of-sample R^2 and daily return correspondingly) performance. We also provided the difference in the daily return of each model compared with FarmPredict. Table 9 shows that FarmPredict outperformed all other models in terms of both prediction accuracy and portfolio performance. SESTM performs the best among all others except FarmPredict (mainly in the last several windows), whereas there is still a 1.2 bps difference compared with FarmPredict.

6. Conclusion

Previous studies on text data usually rely on a predefined dictionary and humans' prior experience, resulting in a nonadaptive and incomplete capture of information. In contrast to these models, we proposed a novel analytical framework for textual studies that conduct unsupervised information extraction: FarmPredict. FarmPredict first isolates the hidden factors and idiosyncratic components as a vector from high-dimensional text data via unsupervised learning without reliance on prior knowledge. Then, we screen the idiosyncratic components according to their correlations with corresponding beta-adjusted returns conditional on hidden factors. This step is optional but helps reduce the computational cost. Even though only a part of the words is selected, all information is used for screening because of embedded factors. In other words, FarmPredict transforms the high-dimensional data into important factors and useful idiosyncratic components; then, it uses them as the input for further penalized

Table 9. Summary of Model Comparison

Model	Embedding	R^2, %	Daily return, bps	Difference in return, bps
FarmPredict	—	4.21	17.8	—
LASSO	—	3.99	6.4	11.2***
MNIR	—	—	5.9	11.9***
SESTM	—	—	16.5	1.2
Textual factor (SVD)	—	1.26	13.5	4.3**
Chinese BERT	Randomized	−0.04	1.3	—
Chinese BERT	Pretrained	0.75	6.4	10.8***
BiRNN	Randomized	−0.06	2.5	—
BiRNN	Pretrained	1.33	8.5	9.2***
TextCNN	Randomized	−0.67	3.3	—
TextCNN	Pretrained	1.20	9.7	8.0***

Notes. This table shows the fitting result (out-of-sample R^2) and the portfolio performance of models. The word lists in all machine learning models are encoded into integer sequences with the BertTokenizer provided by Hugging Face. We separately used randomized initialization and parameters from the pretrained model published by Google (Devlin et al. 2019) for embedding, as shown in column 2. All other components are the same as those in Table 8. As MNIR and SESTM did not directly predict the returns but the sentiment instead, we did not provide the R^2 result. Statistical significance is indicated with asterisks.
$^{**}p < 0.05$; $^{***}p < 0.01$.

regression or other prediction models. FarmPredict alleviates the information loss by the traditional factor regression in dimensionality reduction and ameliorates the model selection inconsistency in the penalized regression (Fan et al. 2020b).

To demonstrate its applicability, we applied FarmPredict on news data to the Chinese stock market to verify our novel framework's effectiveness in several ways. These include analysis of selected words, the correlation between machine-learned sentiments and financial returns, and the returns of sentiment-based portfolios. The results prove that FarmPredict can extract useful information from an article as exemplified by rarely selected words and phrases in previous studies. The empirical results emphasized that the sentiment scores from our model are a powerful predictor in asset pricing and revealed the mechanism of market response to related news. Finally, we used a simple trading strategy on portfolio construction to realize our model's advantage in textual analysis and prediction power, where our accumulated return outperforms other models.

FarmPredict can extract all information from text data by converting correlated high-dimensional data into weakly correlated data in an unsupervised manner. Therefore, not only is it a novel model for financial analysis, but also, FarmPredict is a general and adaptive supervised learning framework for high-dimensional data, like text analysis in this paper, with flexibility in the choice of method in each process.

Acknowledgments

The authors are grateful for various comments and suggestions made by Shuyi Ge, Oliver Linton, Stefan Nagel, Wei Xiong, Dacheng Xiu, and anonymous reviewers among others. The authors also acknowledge the research assistance by Yuan Gao and Danchun Chen.

Endnotes

[1] Early in 1933, Cowles (1933) manually clustered the sentiment of *The Wall Street Journal* for analysis in the stock market.

[2] For more examples of studies using the dictionary-based method, see Tetlock (2007), Tetlock et al. (2008), García (2013), Da et al. (2015), Calomiris and Mamaysky (2019), and Glasserman and Mamaysky (2019).

[3] Beta-adjusted return for stock i on day t is defined as r^*_{it} = Raw Return$(r_{it}) - \beta_i \cdot$ Market Return(r_t^{market}), where β_i describes the linear relationship between market risk and individual asset returns. This beta-adjusted return makes outcomes, such as sentiment learning, less dependent on the market conditions.

[4] For data construction, we used the original structure of the data in this paper, whereas FarmPredict allows for adding interactions of each variable. The screening process in FarmPredict is also optional for computation power reduction. Finally, FarmPredict is also suitable for other machine learning models, such as random forest, boosting trees, etc.

[5] See Section 4.2 for portfolio-building details.

[6] See the online appendix for details.

[7] Unlike alphabet-based languages (phonograms) such as English, Chinese is a character-based language (logogram). Chinese is constructed with stand-alone Chinese characters with clear meanings on their own. The "words" in Chinese can be based on one or multiple characters. Compared with English, words in Chinese are more flexible, and vocabulary can grow quickly over time. As almost every single character is meaningful on its own, a correct segmentation depends highly on the context of each sentence, especially as each word or phrase can take on multiple meanings (Deng et al. 2016).

[8] Here, we refer collectively to both words and phrases as words for simplicity. The median length of articles is 309 words, 209 of them distinctive.

[9] See Bai and Ng (2002), Stock and Watson (2002), and Fan et al. (2020c) for more details.

[10] Fan et al. (2020c) suggest taking $C = 1$, but this is too small for our application. It is well known that the largest eigenvalues are biased upward. The correction is as follows (Bai and Ding 2012). Let $\hat{\lambda}_j$ be empirical eigenvalues and $p = |\mathbf{D}^{\text{freq}}|$ be the dimension. For a given j, define

$$m_{n,j}(z) = (p-j)^{-1}\left[\sum_{\ell=j+1}^{p}(\hat{\lambda}_\ell - z)^{-1} + ((3\hat{\lambda}_j + \hat{\lambda}_{j+1})/4 - z)^{-1}\right],$$

$$\underline{m}_{n,j}(z) = -(1-\rho_{j,n-1})z^{-1} + \rho_{j,n-1}m_{n,j}(z),$$

with $\rho_{j,n-1} = (p-j)/(n-1)$. The corrected eigenvalue of $\hat{\lambda}_j$ is defined as $\hat{\lambda}_j^C = -\frac{1}{\underline{m}_{n,j}(\hat{\lambda}_j)}$. In our application, because n is much larger than p, this step of correction is very small and can be ignored.

[11] The computation can be done expeditiously because $\widehat{\mathbf{B}}^T\widehat{\mathbf{B}}$ is a diagonal matrix, with the diagonal elements being the k-largest eigenvalues of the matrix $\mathbf{X}\mathbf{X}^T/n$.

[12] See Section A.2 in the online appendix. In the case of applying the logistic regression technique, conditional screening (2.5) and conditional prediction (2.7) should be modified accordingly for the logistic regression model; see Fan et al. (2020c).

[13] SSEC is a market value-weighted index of all stocks in the Shanghai Stock Exchange.

[14] Jieba is an open-source Python package for Chinese word segmentation. It is available on GitHub at github.com/fxsjy/jieba/commit/cb0de2973b2fafaa67a0245a14206d8be70db515.

[15] As shown in Table 1, the sample size almost doubled without down sampling. The computing time and memory needed would nearly quadruple.

[16] We also checked the results using the full data without down sampling. They do not change very much or alter our conclusion. See the sensitivity test in Section A.2.5 in the online appendix.

[17] There are fewer data in February and the first weeks of May and October; this corresponds to the three largest holidays in China. The dates for Chinese spring festivals are based on the traditional Chinese calendar and can happen from late January to late February. Labor Day golden week and National Day golden week take place on the first days of May and October, respectively, and each lasts for a whole week.

[18] The portfolio longs the stocks with the top 50 predicted scores and shorts with the 50 lowest with 1% capital each. More details can be found in Section 4.2. The remaining capital will be kept as cash if fewer than 50 stocks are selected.

[19] We also tested the choices of $C = 30$ and $C = 1$ (suggested by Fan et al. 2020c). They result in 80 and 1,043 weak factors, respectively. Because our sample size is very large, the overestimation of k is not a serious problem, and the results are very similar. More details regarding the choices of the number of factors can be found in Section A.2.2 in the online appendix.

[20] The positive and negative words in the U.S. market are cited from Ke et al. (2019). The positive words include undervalue, repurchase, surpass, upgrade, and rally, and the negative words are shortfall, downgrade, disappointing, tumble, and blame.

[21] We chose $t-5$, which corresponds to past one-week lagged returns. This mitigates the days of the week effect.

[22] If there are multiple articles of stock during the same day, we separately estimated their sentiment scores and then averaged them as the final sentiment score.

[23] The average and standard deviation are used for a quick summary of the distribution of the sentiments of daily news articles. They can be replaced by quintiles or deciles for a more informative summary.

[24] We assume that the latest news will have a higher power to affect beta-adjusted returns of stocks. Hence, if other news occurred within the 14-day range of the former news, we will recalculate and renew the periods of the day indicator.

[25] We further tested the performance with a different number of stocks. See the details in the online appendix.

[26] For special treatment stocks, the limit is 5%.

[27] Such a mechanism might affect price discovery in several ways (Chen et al. 2019). On one hand, stock prices failing to reach their fair values because of the limit might continue to move in the same direction the next day. On the other hand, it is widely believed by the Chinese media and Chinese investors that some limits are artificially hit by speculators for price manipulation purposes to lure people to buy and that prices will revert the next day.

[28] As we noted, although these stocks might not be able to trade because of the price limits, with higher-frequency data, the portfolio would reach better performances.

[29] Note that the R^2_{factor} is not a result of OLS regression. It is borrowed from OLS's definitions to illustrate the number of market movements in our portfolio.

[30] Here, the factor-adjusted returns are estimated by $R_{p,t} - R_f - \beta_1\text{MKT}_t - \beta_2\text{SMB}_t - \beta_3\text{HML}_t - \beta_4\text{MOM}_t$.

[31] We combined the 10 test durations as one and calculated the total R^2 of the model. Daily returns are calculated using the EW portfolio-building method, as mentioned in Section 4.2.

[32] The same κ is used for comparison; hence, the comparison is conducted under the same dimension. We trained the LASSO model following the same tuning process of FarmPredict: that is, the hyperparameter λ for penalization is tuned using the data from 2000 to 2014 and then fixed. We only estimated coefficients β in the rolling windows.

[33] This thought is consistent with the double-robust properties in Arkhangelsky et al. (2021).

[34] The function words are removed based on the HIT stop words. See https://github.com/goto456/stopwords for details.

[35] See https://cran.r-project.org/web/packages/textir/index.html. We tuned the hyperparameters gamma and nlambda in the package following the same process of FarmPredict.

[36] The Chinese word2vec model can be found at https://github.com/Embedding/Chinese-Word-Vectors. It is not necessary to use a pretrained word2vec model rather than training one by the current data. However, because the training of word2vec models would cost lots of computational resources and data, pretrained models are widely used.

[37] See the online appendix for more details.

[38] The duration may vary because of the different data structure and festival of each month.

[39] See Schuster and Paliwal (1997) and Chen (2015) for model details.

[40] See Table 9 for detailed fitting and portfolio performance results.

References

Ahn SC, Horenstein AR (2013) Eigenvalue ratio test for the number of factors. *Econometrica* 81(3):1203–1227.

Antweiler W, Frank MZ (2004) Is all that talk just noise? The information content of internet stock message boards. *J. Finance* 59(3): 1259–1294.

Arkhangelsky D, Athey S, Hirshberg DA, Imbens GW, Wager S (2021) Synthetic difference-in-differences. *Amer. Econom. Rev.* 111(12): 4088–4118.

Bai Z, Ding X (2012) Estimation of spiked eigenvalues in spiked models. *Random Matrices Theory Appl.* 1(02):1–21.

Bai J, Ng S (2002) Determining the number of factors in approximate factor models. *Econometrica* 70(1):191–221.

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. *J. Machine Learn. Res.* 3(January):993–1022.

Calomiris CW, Mamaysky H (2019) How news and its context drive risk and returns around the world. *J. Financial Econom.* 133(2):299–336.

Carhart MM (1997) On persistence in mutual fund performance. *J. Finance* 52(1):57–82.

Chen Y (2015) Convolutional neural network for sentence classification. *UWSpace* (August 26), https://uwspace.uwaterloo.ca/handle/10012/9592.

Chen J, Jiang F, Tu J (2015) Asset allocation in the Chinese stock market: The role of return predictability. *J. Portfolio Management* 41(5):71–83.

Chen T, Gao Z, He J, Jiang W, Xiong W (2019) Daily price limits and destructive market behavior. *J. Econometrics* 208(1):249–264.

Cong LW, Liang T, Zhang X (2019) Textual factors: A scalable, interpretable, and data-driven approach to analyzing unstructured information. Preprint, submitted September 1, https://dx.doi.org/10.2139/ssrn.3307057.

Cowles A (1933) Can stock market forecasters forecast? *Econometrica* 1(3):309–324.

Da Z, Engelberg J, Gao P (2015) The sum of all FEARS investor sentiment and asset prices. *Rev. Financial Stud.* 28(1):1–32.

Deng K, Bol PK, Li KJ, Liu JS (2016) On the unsupervised analysis of domain-specific Chinese texts. *Proc. Natl. Acad. Sci. USA* 113(22):6154–6159.

Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. Preprint, submitted May 24, https://arxiv.org/abs/1810.04805.

Du Z, Huang AG, Wermers R, Wu W (2022) Language and domain specificity: A Chinese financial sentiment dictionary. *Rev. Finance* 26(3):673–719.

Fama EF, French KR (1993) Common risk factors in the returns on stocks and bonds. *J. Financial Econom.* 33(1):3–56.

Fan J, Lv J (2008) Sure independence screening for ultrahigh dimensional feature space. *J. Roy. Statist. Soc. Ser. B Statist. Methodology* 70(5):849–911.

Fan J, Guo J, Zheng S (2020a) Estimating number of factors by adjusted eigenvalues thresholding. *J. Amer. Statist. Assoc.* 117(538):852–861.

Fan J, Ke Y, Wang K (2020b) Factor-adjusted regularized model selection. *J. Econometrics* 216(1):71–85.

Fan J, Li R, Zhang C-H, Zou H (2020c) *Statistical Foundations of Data Science* (CRC Press, Boca Raton, FL).

Gao Z, Ren H, Zhang B (2020) Googling investor sentiment around the world. *J. Financial Quant. Anal.* 55(2):549–580.

García D (2013) Sentiment during recessions. *J. Finance* 68(3):1267–1300.

Gentzkow M, Kelly B, Taddy M (2019a) Text as data. *J. Econom. Literature* 57(3):535–574.

Gentzkow M, Shapiro JM, Taddy M (2019b) Measuring group differences in high-dimensional choices: Method and application to congressional speech. *Econometrica* 87(4):1307–1340.

Glasserman P, Mamaysky H (2019) Does unusual news forecast market stress? *J. Financial Quant. Anal.* 54(5):1937–1974.

Gu S, Kelly B, Xiu D (2020) Empirical asset pricing via machine learning. *Rev. Financial Stud.* 33(5):2223–2273.

Henry E (1973) Are investors influenced by how earnings press releases are written? *J. Bus. Comm.* 45(4):363–407.

Horel E, Giesecke K (2020) Significance tests for neural networks. *J. Machine Learn. Res.* 21(227):1–29.

Jegadeesh N, Wu D (2013) Word power: A new approach for content analysis. *J. Financial Econom.* 110(3):712–729.

Ke ZT, Kelly BT, Xiu D (2019) Predicting returns with text data. NBER Working Paper No. 26186, National Bureau of Economic Research, Cambridge, MA.

Larsen V, Thorsrud LA (2017) Asset returns, news topics, and media effects. Preprint, submitted September 19, https://dx.doi.org/10.2139/ssrn.3057950.

Loughran T, McDonald B (2011) When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *J. Finance* 66(1):35–65.

Loughran T, McDonald B (2016) Textual analysis in accounting and finance: A survey. *J. Accounting Res.* 54(4):1187–1230.

Manela A, Moreira A (2017) News implied volatility and disaster concerns. *J. Financial Econom.* 123(1):137–162.

Nagel S (2005) Short sales, institutional investors and the cross-section of stock returns. *J. Financial Econom.* 78(2):277–309.

Nagel S (2021) *Machine Learning in Asset Pricing* (Princeton University Press, Princeton, NJ).

Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing* 45(11):2673–2681.

Stock JH, Watson MW (2002) Forecasting using principal components from a large number of predictors. *J. Amer. Statist. Assoc.* 97(460):1167–1179.

Sun J (2017) Jieba Version v0.39 (August 31). https://github.com/fxsjy/jieba.

Sun L, Najand M, Shen J (2016) Stock return predictability and investor sentiment: A high-frequency perspective. *J. Banking Finance* 73(11):147–164.

Taddy M (2013) Multinomial inverse regression for text analysis. *J. Amer. Statist. Assoc.* 108(503):755–770.

Tetlock PC (2007) Giving content to investor sentiment: The role of media in the stock market. *J. Finance* 62(3):1139–1168.

Tetlock PC, Saar-Tsechansky M, Macskassy S (2008) More than words: Quantifying language to measure firms' fundamentals. *J. Finance* 63(3):1437–1467.