How Much Can Machines Learn Finance from Chinese Text Data?

Yang Zhou, Jianqing Fan, Lirong Xue (MS, 2024)

Present by Li Ziming

Motivation

- Limitations of existing textual analysis methods.
  - **Dictionary approach**: rely on predefined dictionary; introduce subjectivity and bias; lack adaptability across markets and time periods.
  - **Word selection by text regression**: only capture single-word correlation; prone to overfitting; overlook holistic semantics.
  - **Generative topic model (LDA)**: rely on prior knowledge and statistical assumption; provide unstable result; limit generalizability.
  - **Cluster words into topic (BERT & word2vec)**: pretraining not designed for financial prediction; high-dimensional embedding introduce noise.

New framework for learning textual data

- **FarmPredict**: factor-augmented regularized model
  - **Generalizability**: without predefined knowledge; adapt to different tasks.
  - **Robustness**: extract information directly from article; avoid subjective bias.
  - **Adaptability**: automatically balance between word selecting and clustering; reduce information loss in dimensionality reduction.
  - **Prediction enhancement**: combine effects of hidden topics and individual words; enrich predictor variables.

Contribution

- Contribute to literature on financial textual analysis.
  - Prior literature: dictionary (Loughran and McDonald, 2016); word selection by text regression (Gentzkow et al., 2019); LDA (Ke et al., 2019); word2vec (Cong et al., 2019).
  - Extend: introduce a novel factor-augmented regularized model for prediction.
- Contribute to study of emerging markets in Chinese context.
  - Prior literature: most conducted under language environment in **English** and relatively **developed** financial markets (Calomiris and Mamaysky, 2019).
  - Extend: show possibility of applications of machine learning techniques in languages other than English and developing markets.

Problem setup

- Article representation: word set
  - Assumption1: word-level statistics can summarize all information in the article.
  - D: entire set of all possible Chinese words in n articles.
  - $d_i \in N^{|D|}$: vector of word counts of every word in the ith article.
- Data characteristics: high-dimensional and sparse
  - 914,000 articles, 1,181,000 distinctive words, 71,000 words appear in at least 50 articles.
- Word classification
  - Assumption2: predict target mainly affected by small subset of words.
  - All words divided into two disjoint categories (set of sentiment-charged words S and set of sentiment-neutral words N, $D = S \cup N$).

FarmPredict

- Step1: selecting frequent words
  - Filter out infrequent words which unlikely to appear in new articles to be scored.
  - $D^{freq} = \{jth\ word\ in\ D\colon\ k_j \geq \kappa\}$
  - Threshold κ tuned as hyperparameter to strike balance between **comprehensiveness** of D^{freq} and **noise** introduced by infrequent words.

FarmPredict

- Step2: factor analysis
  - X_i: feature vector of ith article, which $X_{i,j}$ is feature of word $j \in D^{freq}$ in ith article.
  - Assumption3: dependence among words assumed driven by latent factors
  - $X_i = Bf_i + u_i,\ n = 1, \ldots, n$ (Matrix form $X = FB^{\intercal} + U$)
  - F: $n \times k$ matrix of latent factors, topic or low-dimensional representation of article
  - B: $\left|D^{freq}\right| \times k$ loading matrix, extract relationship between each word and factors (which words belong to which factors)
  - U: n $\times \left|D^{freq}\right|$ matrix of idiosyncratic components, cannot be explained by topics.

FarmPredict

- Step3: predict modeling
  - Correlated features in X_i disentangled into factors f_i and idiosyncratic components u_i.
  - X_i to predict associated return outcome Y_i, use f_i and u_i as predictor.
  - $Y_i = a + b^\intercal f_i + \beta^\intercal u_i + \epsilon_i$
  - Linear space spanned by X_i same as that spanned by f_i and u_i, but use f_i and u_i **augment predictors** by latent factors and variables **less correlated**.

FarmPredict

- Step4: learning factors and idiosyncratic components
  - Given number of factors k, use PCA to estimate $X = FB^{\intercal} + U$
  - Eigen-decomposition: $XX^{\intercal} = V\Lambda V^{\intercal}$ (V: eigenvectors; Λ: eigenvalues matrix)
  - Latent factor $\hat{F} = \sqrt{n}V_k$ (eigenvectors of largest k eigenvalues)
  - Loading matrix $\hat{B} = X^{\intercal}\hat{F}/n$
  - Idiosyncratic components $\hat{U} = X - \hat{F}\hat{B}^{\intercal}$
  - Select number of factors k: adjusted eigenvalue thresholding (Fan et al., 2020)
  - $\hat{k} = \max\{j < |D^{freq}| : \hat{\lambda}_j^{\,c} > 1 + C\sqrt{|D^{freq}|/(n-1)}\}$

FarmPredict

- Step5: learning conditional sentiment-charged words S
  - Remove factor effect and get residual: $\widehat{Y_u} = Y - \hat{Y}(\hat{F})$
  - Part that cannot explained by factor is predicted by sentiment-charged components.
  - Screen words by **return-relationship** and **appear frequency**.
  - $\hat{S} = \{j: |corr(\widehat{U_j}, \widehat{Y_u}) > \alpha|\} \cap \{j: k_j > \kappa\}$
  - Sentiment-charged component sub-vector $u_{i,\hat{S}} = (u_{i,j}: j \in \hat{S})$

FarmPredict

- Step6: fitting FarmPredict
  - Penalized least squares regression
  - $\hat{a},\ \hat{b},\ \hat{\beta} = argmin_{a,\,b,\,\beta}\{\frac{1}{n}\sum_i (Y_i - a - b^{\intercal} f_i - \beta^{\intercal} u_{i,\hat{S}})^2 + \lambda\|\beta\|_1 + \lambda\|b\|_1\}$
  - Penalty parameter λ chosen by cross validation, control model bias-variance trade-off and sparsity of $\widehat{b}$ and $\hat{\beta}$.
  - Further reduce sentiment-charged words.

FarmPredict

- Step7: scoring new articles
  - Feature vector of new article X_{new} decomposed with well-trained $\hat{B}$.
  - Least squares optimization problem: $\underset{f_{new}}{argmin}(X_{new} - \hat{B}f_{new})^2$
  - $\frac{\partial}{\partial f_{new}}\left(X_{new} - \hat{B}f_{new}\right)^2 = -2\hat{B}^\intercal\left(X_{new} - \hat{B}f_{new}\right), \hat{B}^\intercal X_{new} - \hat{B}^\intercal\hat{B}f_{new} = 0$
  - Latent factor $f_{new} = (\hat{B}^\intercal\hat{B})^{-1}\hat{B}^\intercal X_{new}$; idiosyncratic component $u_{new} = X_{new} - \hat{B}\widehat{f_{new}}$
  - **Sentiment score** $\widehat{Y_{new}} = \hat{a} + \hat{b}^\intercal f_{new} + \hat{\beta}^\intercal u_{new,\hat{S}}$
  - Multiple articles of stock in same day, separately estimate and average as final score.

Data

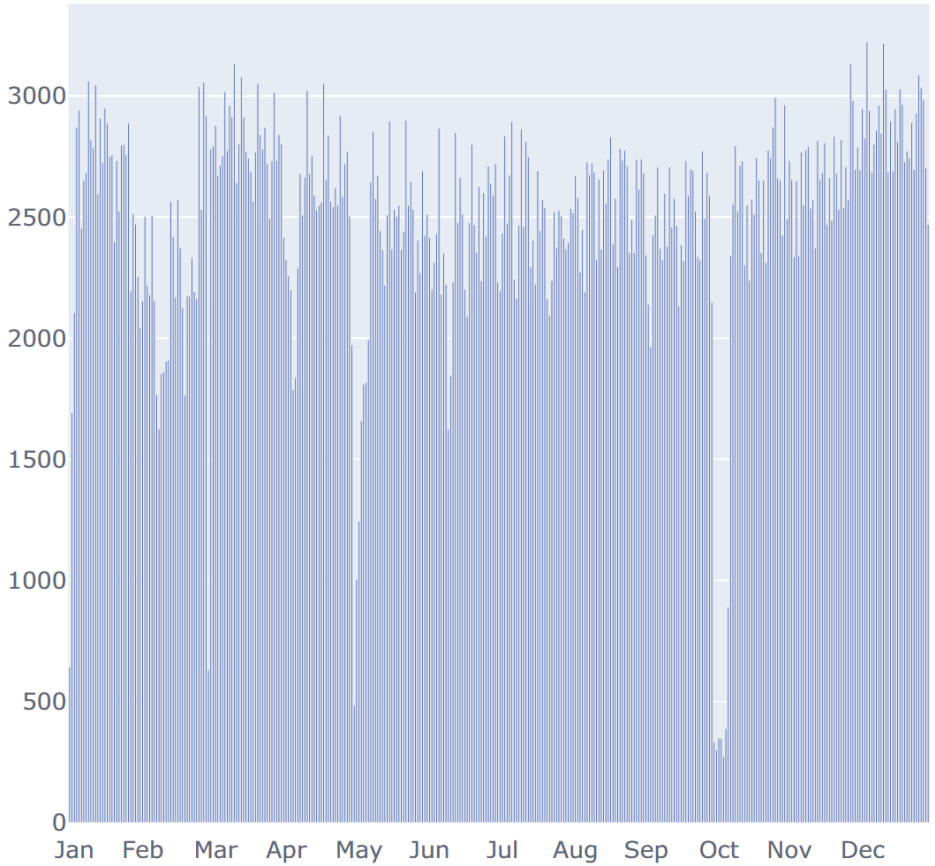
- Data collection: Sina Finance
  - Scrawling in **Breadth-First Search**: Start with main page; download HTML files get all links in it; filter links belong to sina.com.cn domain; add to queue for subsequent access; repeat above process until queue empty. (visit 6.3 million links, 5.8 million valid)
- Preprocessing
  - Deduplication: remove same title article published on same day.
  - Cleaning: trim to **Chinese** character, all html digit and punctuation mark stripped away.
  - Return matching: **close-to-close return** covering article publish time (Beta-adjusted)
  - Word segmentation: Jieba (based on hidden Markov model)
  - Down sampling: randomly sample 300 articles at most each day (evenly distributed)

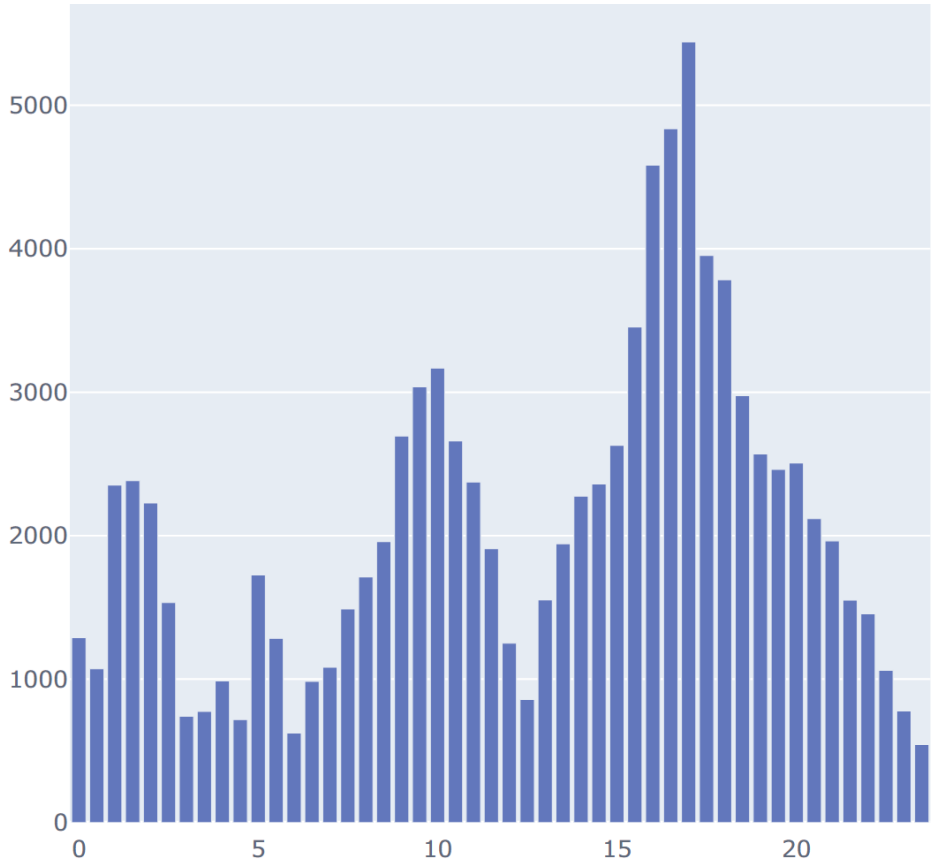
Basic statistics

- Data scale
  - 914,000 articles, 1,181,000 distinctive words, 71,000 words appear in at least 50 articles.
- Sparse feature
  - Each article have a median of 376 words and 209 distinct words.
  - 0.29% is nonzero entries among 71,000 dimensional vectors.

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%

- Time distribution
  - Daily distribution: even across each day except for a couple of holidays
  - Intraday distribution: most from market open time to end of day.





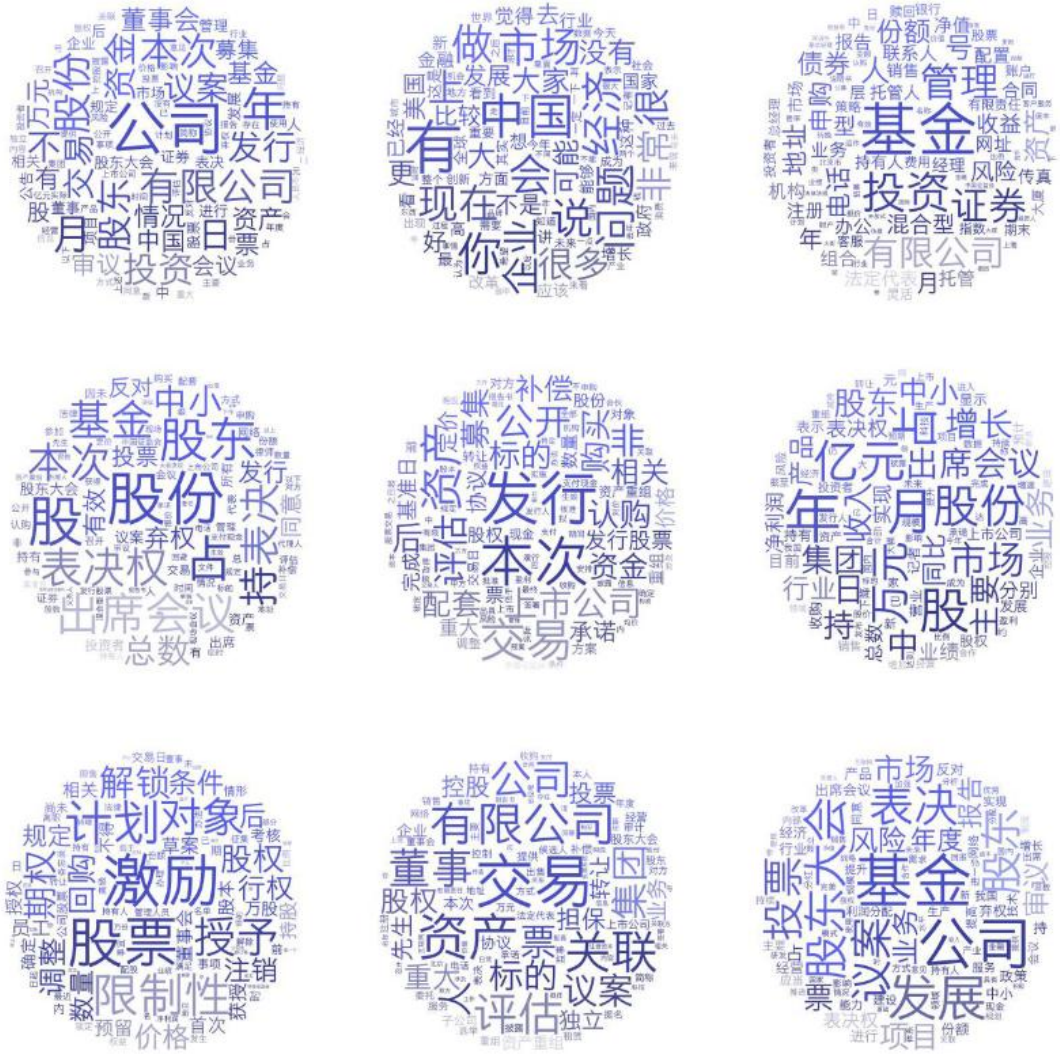
Tuning and testing

- Tuning hyperparameters

Hyperparameters	Range	Optimal Parameter
C	Choose from 1 to 200	150(k=9)
κ	Choose from 80% to 96% with increments of 2%	\
α	Control number of words in $\lvert\hat{S}\rvert$ (500/1,000/ 2,000)	\
λ	Ensure words with non-zero coefficients (20/50/100/200)	\

- Rolling windows test
  - Window length: 10 years of training data + 6 months of testing data
  - Rolling procedure: move window forward by 6 months each time
  - Total windows: 10, covering all testing periods from 2015 to 2019

Content of factors



Top sentiment-charged words

- Words in $\widehat{S}$ and ranked by $\hat{\beta}$ as sentiment strength

	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

Do sentiments predict returns?

	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

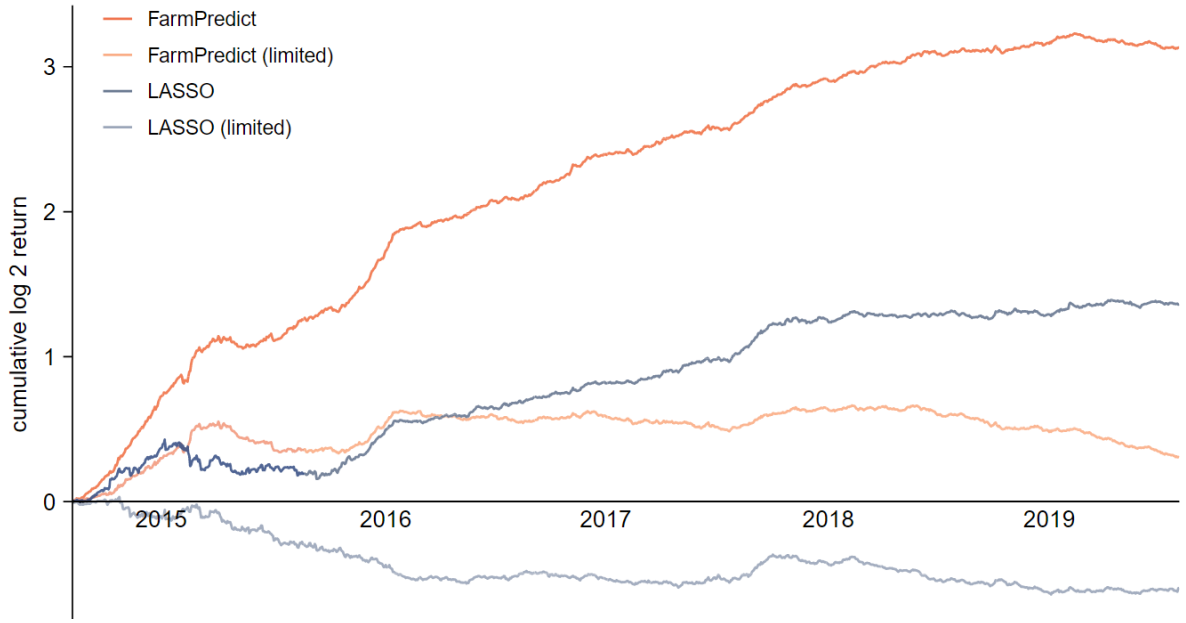
Return	(1)	(2)	(3)	(4)	(5)	(6)
$AveSentiment_{t-1}$	0.006	0.053	0.004	−0.004	−0.031	−0.045
	(0.118)	(0.125)	(0.129)	(0.131)	(0.134)	(0.144)
$DISP_{t-1}$		−0.251	−0.286	−0.310	−0.187	−0.132
		(0.226)	(0.228)	(0.228)	(0.238)	(0.261)
Market variables			Yes	Yes	Yes	Yes
Short horizon lagged return				Yes	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

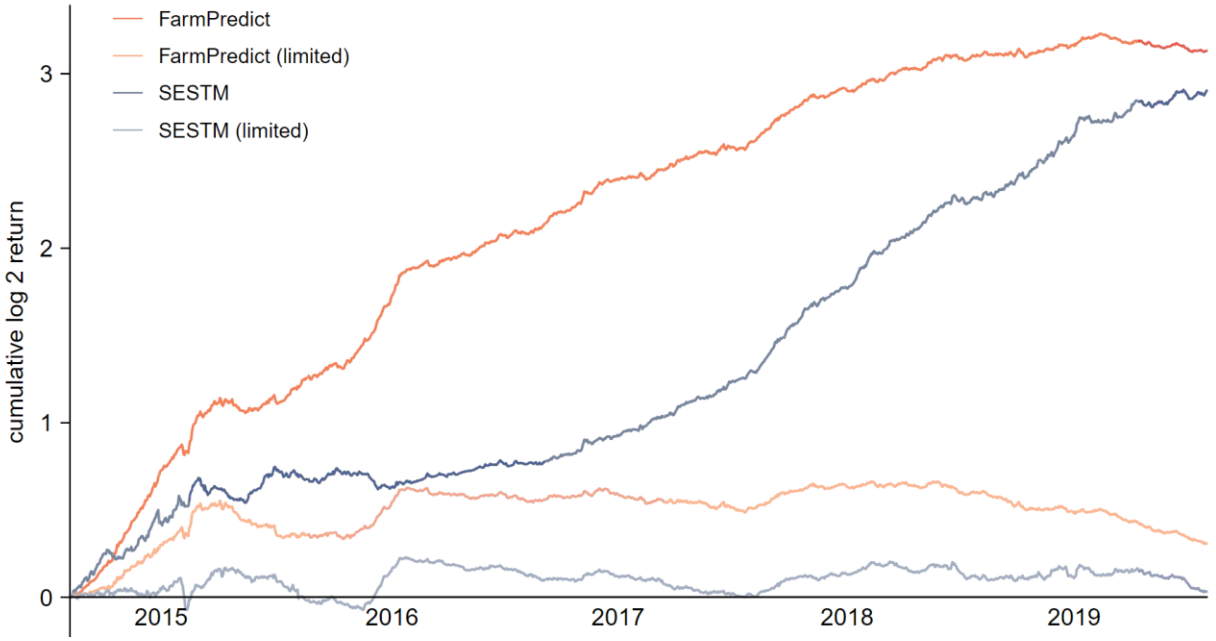
Portfolio performance

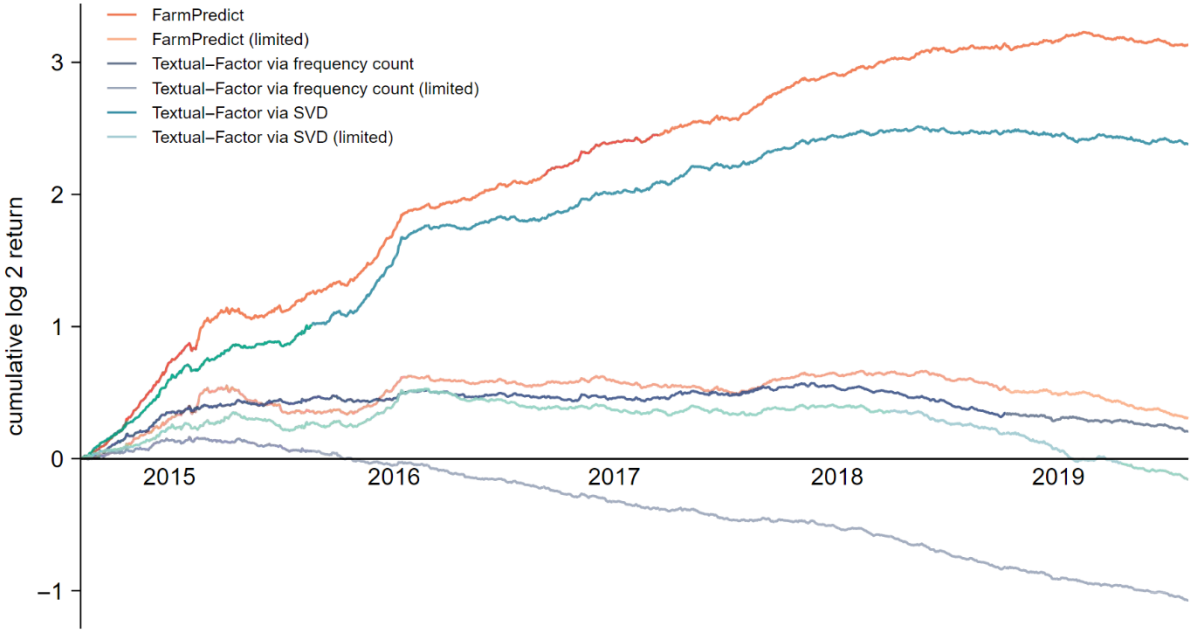
	Upper bound (without price limits)				Lower bound (with price limits)			
	No Transac		With Transac		No Transac		With Transac	
Portfolio	SR	APR, %	SR	APR, %	SR	APR, %	SR	APR, %
EW								
L + S	8.51	105.16	5.32	54.35	4.26	38.70	0.21	4.30
L	3.86	75.76	4.36	52.44	1.74	20.18	0.17	4.21
S	1.14	16.30	−0.18	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

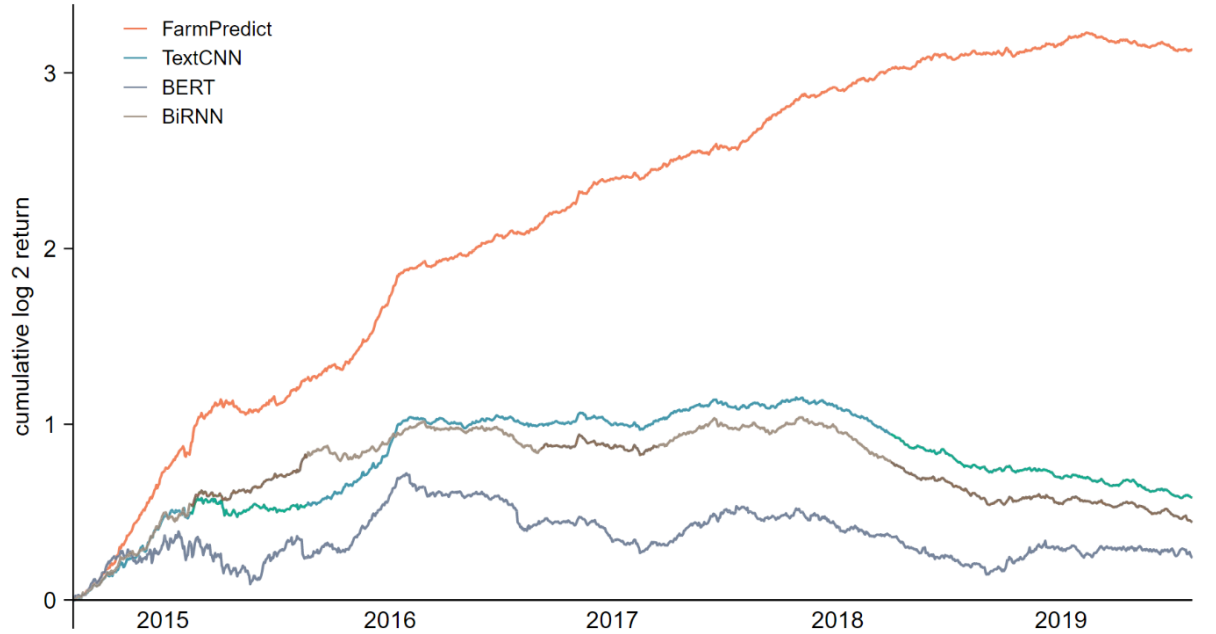
Model comparison

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***


FarmPredict
FarmPredict (limited)
LASSO
LASSO (limited)
cumulative log 2 return
3
2
1
0
2015
2016
2017
2018
2019


FarmPredict
FarmPredict (limited)
SESTM
SESTM (limited)
cumulative log 2 return
3
2
1
0
2015
2016
2017
2018
2019


FarmPredict
FarmPredict (limited)
Textual–Factor via frequency count
Textual–Factor via frequency count (limited)
Textual–Factor via SVD
Textual–Factor via SVD (limited)
cumulative log 2 return
3
2
1
0
−1
2015
2016
2017
2018
2019


FarmPredict
TextCNN
BERT
BiRNN
cumulative log 2 return
3
2
1
0
2015
2016
2017
2018
2019

New ideas

- Change predicted target: (positive/negative) earning changes; fraud detection; corporate actions.
- Replace linear prediction model by other ML model like decision tree or neural networks
- How to combine digital information in article