Predicting Returns With Text Data

Tracy Ke
Harvard University

Bryan Kelly
Yale University

Dacheng Xiu
University of Chicago

36th Canadian Econometric Study Group Meetings
October 2019

Machine Learning and Finance

Can Machines Learn Finance?

- "Will AI-Powered Hedge Funds Outsmart the Market?" **MIT Technology Review**

  "... One of the most promising uses of relatively new AI techniques may be processing unstructured natural language data in the form of news articles, company reports, and social media posts, in an effort to glean insights into the future performance of companies, currencies, commodities, or financial instruments. ... " *February 4, 2016*

Why AI?

- The numerical representation of text as data for statistical analysis is, in principle, ultra-high dimensional
- Machine learning offers a toolkit for tackling the high-dimensional statistical problem of extracting meaning from text for explanatory and predictive analysis

Textural Analysis in Finance

NLP/ML increasingly sophisticated for modeling complexities of verbal communication

- AI-focused fund: Sentient, Rebellion Research, AIEQ, ...
- RavenPack, Refinitiv, Bloomberg, ...

Common approach: deep neural networks

- "Trading, Evolved." **Source:** `http://www.sentientim.com/about/`

  Combining evolutionary intelligence technologies, deep learning algorithms and other techniques that identify and propagate the most successful strategies, SIM's distributed artificial intelligence system is continuously processing and learning from enormous stockpiles of data, developing investment strategies in groundbreaking new ways.

Textural Analysis in Finance

NLP/ML increasingly sophisticated for modeling complexities of verbal communication

- AI-focused fund: Sentient, Rebellion Research, AIEQ, ...
- RavenPack, Refinitiv, Bloomberg, ...

Common approach: deep neural networks

- "Trading, Evolved." **Source:** `http://www.sentientim.com/about/`

  Combining evolutionary intelligence technologies, deep learning algorithms and other techniques that identify and propagate the most successful strategies, SIM's distributed artificial intelligence system is continuously processing and learning from enormous stockpiles of data, developing investment strategies in groundbreaking new ways.

Performance?

- "AI Hedge Fund Is Said to Liquidate After Less Than Two Years" **Bloomberg**, Sept 6, 2018

Textural Analysis in Finance

But, usage of textual analysis in academic finance still in its infancy

- Most commonly used to study **"sentiment"** of a document

Common approach: Weight terms based on pre-specified sentiment dictionary

- Tetlock (2007): Harvard-IV psychosocial dictionary
- Loughran and McDonald (2007): Create a new dictionary for finance context
- Loughran and McDonald (2011): Aggregate sentiment scores of words via an ad-hoc Term Frequency-Inverse Document Frequency weighting scheme

Sentiment scores then used in secondary statistical model

- E.g., "How do asset returns associate with media sentiment?"

Our Goal - Get more out of the data

ML to understand sentimental structure of text

1. Statistical benchmark model for "supervised" sentiment extraction
   - Model how news are generated and how sentiment is embeded
   - Consider that different contexts demand different notions of sentiment (i.e., supervised)
   - Emphasis on ease of use and transparency (i.e., not just for machine learners)

2. Empirical evidence of gains
   - Translation of statistical gains into economic terms via portfolio performance

Literature

- Most prior work using text as data for finance and accounting research does little, if any, direct statistical analysis of text
  - Tetlock (2007), Loughran and McDonald (2007), Loughran and McDonald (2011)
  - ⋆ We develop a machine learning method to build context-specific sentiment scores

- A few exceptions in the finance literature use machine learning to analyze text
  - Naïve Bayes: Antweiler and Frank (2005), Li (2010), Jegadeesh and Wu (2013), etc.
  - Support Vector Regression: Manela and Moreira (2017)
  - Word2vec, LDA: a few papers at SoFiE this year
  - ⋆ Our model is generative, transparent, tractable, and accompanied by theoretical guarantees

- One prong in broader agenda that embraces complexity in finance
  - Return Prediction: Gu, Kelly, and Xiu (2018)
  - ⋆ This paper addresses the same prediction problem but use alternative data (signals) — news text, as opposed to accounting, fundamental, and price volume technical signals

Model Summary

Data Structure

- Vocabulary of m words: $\{1, 2, \ldots, m\} = S \cup N$.
- Events/documents $i = 1, ..., n$ with word counts $\underbrace{d_i}_{m \times 1}$ with corresponding return r_i

Return Probability Structure

- Each event i has a "sentiment" value, p_i
- $\text{Prob}(r_i > 0 | p_i) = g(p_i)$, with g a strictly increasing function

Text Probability Structure

$$d_{i,[S]} \sim \text{Multinomial}(s_i, p_i O_+ + [1 - p_i] O_-)$$

- O_+ a $|S| \times 1$ vector of expected word frequency in a purely positive article (i.e., when $p_i = 1$)
- O_- a $|S| \times 1$ vector of expected word frequency in a purely negative article (i.e., when $p_i = 0$)
- Each i's mixture proportion determined by p_i

The IBM Example: Raw Article

IBM Profit Falls as Revenue Declines – 4th Update By Robert McMillan

International Business Machines Corp. is trying to reinvent itself as a modern technology innovator, but it is proving to be a tough act for the century-old company.

On Monday, IBM reported second-quarter revenue fell 13.5%, adding to a string of quarterly declines that now spans 13 periods despite scaling back on legacy hardware and pushing into cloud-based software and services.

IBM remains under assault from computing in the cloud, which threatens to undermine its hardware and infrastructure businesses and erode profit margins in the computing business. To win this fight, the company trimmed itself over the past year, exiting unprofitable server and chip-making businesses to focus instead on data analytics and security software as well as cloud and mobile computing products. ... IBM says that these newer businesses are growing, but the company reported a year-over-year decline in all of its major lines. Technology services revenue was down 10%; business services fell 12%; software dropped 10%; and overall hardware revenue sank 32%. IBM profit dipped 16.6% to $3.45 billion, weighed down by acquisition-related charges.

Tess Stynes contributed to this article.

Write to Tess Stynes at tess.stynes@wsj.com and Robert McMillan at Robert.Mcmillan@wsj.com

Access Investor Kit for International Business Machines Corp.

Visit `http://www.companyspotlight.com/partner?cp_code=P479&isin=US4592001014`.

July 20, 2015 19:06 ET (23:06 GMT)

How Machine Reads the IBM News ...

'ibm', 'profit', 'fall', 'revenue', 'decline', 'update'

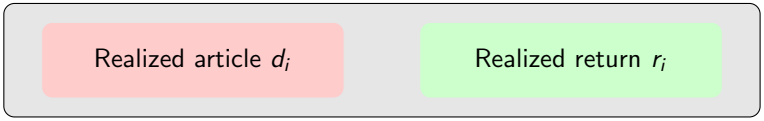
'by', 'try', 'reinvent', 'technology', 'innovator', 'prove', 'tough', 'act', 'century', 'old', 'company', 'on', 'reported', 'second', 'quarter', 'revenue', 'fall', 'add', 'string', 'quarterly', 'decline', 'span', 'period', 'despite', 'scale', 'back', 'legacy', 'push', 'cloud', 'base', 'software', 'service', 'remain', 'assault', 'compute', 'cloud', 'threatens', 'undermine', 'infrastructure', 'business', 'erode', 'profit', 'margin', 'compute', 'business', 'to', 'win', 'fight', 'company', 'trim', 'past', 'year', 'exit', 'unprofitable', 'chip', 'making', 'business', 'focus', 'instead', 'analytics', 'security', 'software', 'well', 'cloud', 'compute', 'product', 'say', 'new', 'business', 'grow', 'company', 'report', 'year', 'year', 'decline', 'major', 'line', 'service', 'revenue', 'business', 'service', 'fall', 'software', 'drop', 'overall', 'revenue', 'sank', 'and', 'worryingly', 'profit', 'margin', 'service', 'software', 'business', 'appear', 'shrink', 'say', 'analyst', 'always', 'move', 'high', 'value', 'snapshot', 'show', 'trouble', 'move', 'margin', 'say', 'low', 'expect', 'tax', 'bill', 'small', 'restructuring', 'cost', 'help', 'company', 'aposs', 'profit', 'soften', 'underperformance', 'core', 'business', 'say', 'result', 'pretty', 'much', 'line', 'expect', 'say', 'dropped', 'hour', 'trade', 'company', 'say', 'cloud', 'compute', 'revenue', 'rise', 'year', 'ago',

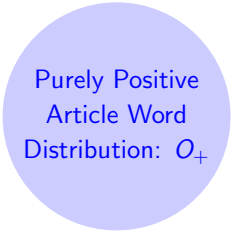
...

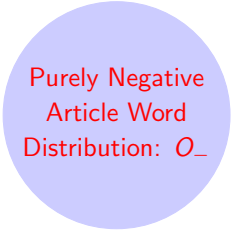
The IBM Example: A Bag of Words Representation

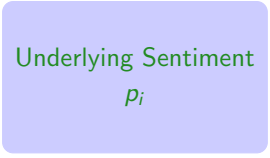
say	11	focus	2	article	1	despite	1	increase	1	much	1	result	1	technology	1
business	9	go	2	assault	1	dip	1	independent	1	necessarily	1	rise	1	tend	1
revenue	9	help	2	back	1	divestiture	1	industry	1	number	1	sale	1	to	1
company	7	move	2	bank	1	division	1	infrastructure	1	on	1	sank	1	tough	1
service	7	new	2	base	1	do	1	innovator	1	overall	1	scale	1	trade	1
cloud	5	old	2	because	1	dollar	1	instead	1	percentage	1	security	1	transaction	1
profit	5	past	2	bill	1	drop	1	insurance	1	period	1	sell	1	transition	1
year	5	product	2	build	1	erode	1	interview	1	platform	1	show	1	trim	1
billion	4	quarter	2	but	1	estimate	1	introduction	1	pretty	1	shrink	1	trouble	1
compute	4	second	2	by	1	exclude	1	investor	1	prove	1	small	1	try	1
fall	4	acquisition	1	century	1	exit	1	job	1	push	1	snapshot	1	undermine	1
line	4	act	1	change	1	expectation	1	jury	1	quarterly	1	soften	1	underpin	1
margin	3	add	1	charge	1	fight	1	keep	1	question	1	span	1	unprofitable	1
still	3	age	1	chip	1	get	1	lately	1	refresh	1	spur	1	use	1
account	2	always	1	computer	1	grow	1	legacy	1	reinvent	1	storage	1	value	1
ago	2	analytics	1	contribute	1	high	1	low	1	relate	1	string	1	weighed	1
analyst	2	and	1	core	1	hour	1	major	1	relevant	1	strong	1	well	1
boost	2	anywhere	1	cost	1	hurt	1	making	1	remain	1	struck	1	widely	1
decline	2	appear	1	currency	1	idea	1	masterful	1	remarkable	1	successfully	1	win	1
expect	2	application	1	deal	1	important	1	month	1	report	1	tax	1	worryingly	1

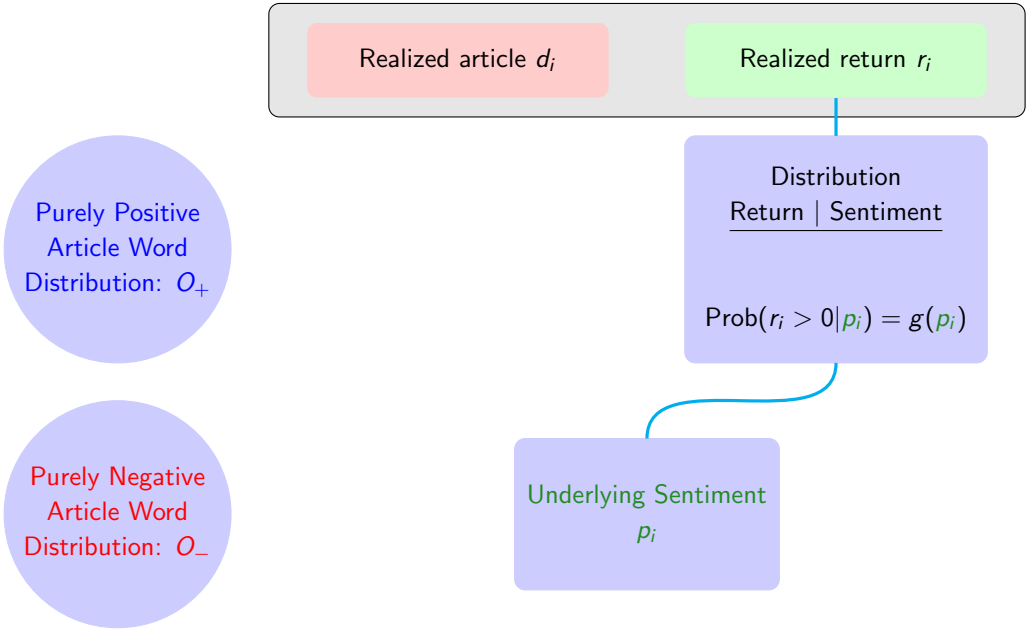
In total, there are 38,862 words in the dictionary, only 160 of which appear in this article.

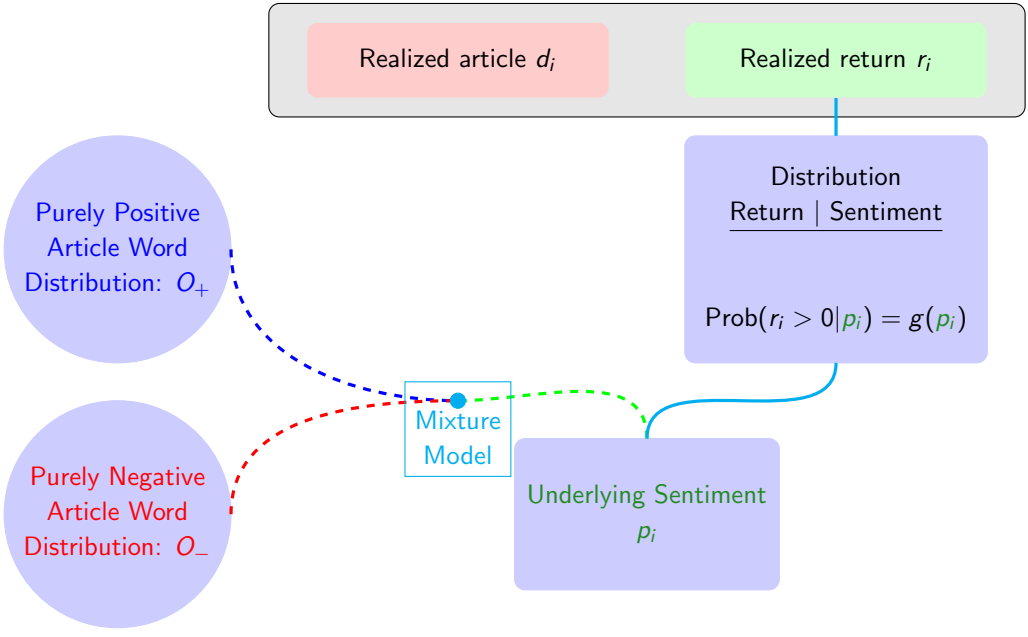

Realized article d_i
Realized return r_i

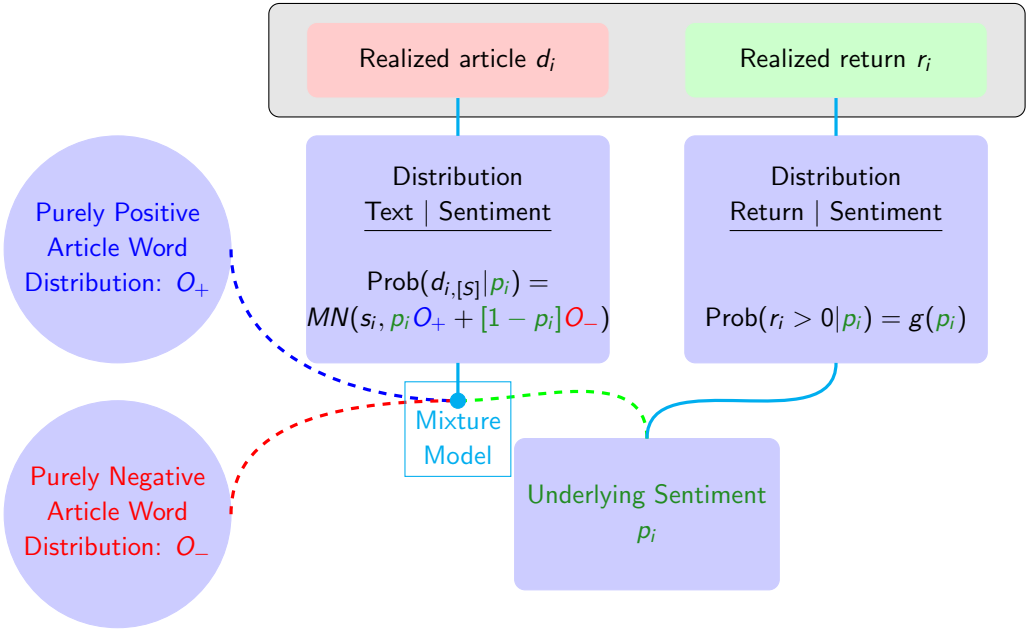

Purely Positive
Article Word
Distribution: O_+

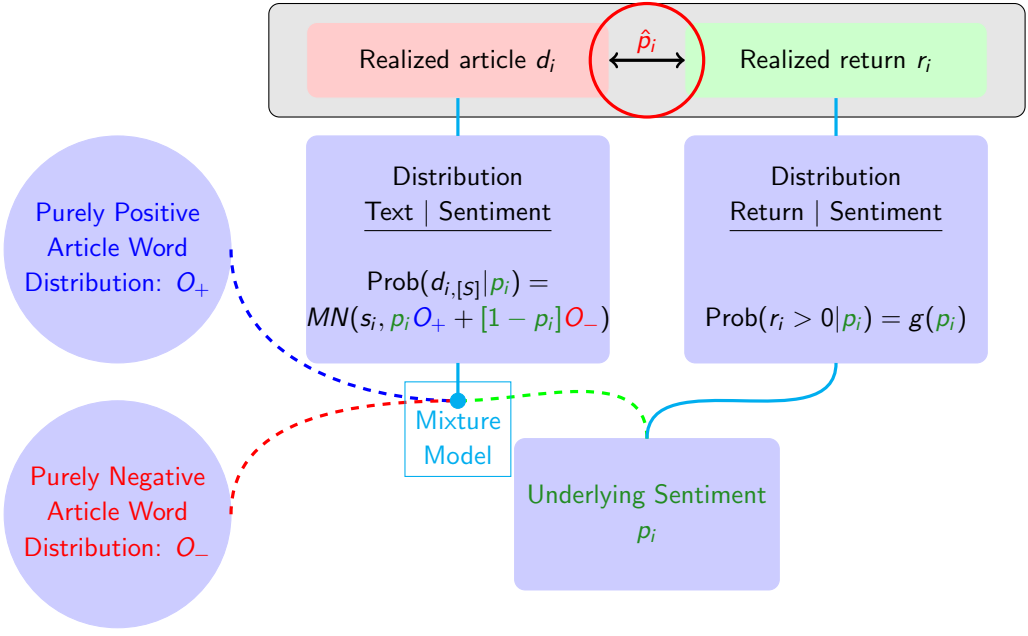

Purely Negative
Article Word
Distribution: O_-


Underlying Sentiment
p_i


Realized article d_i
Realized return r_i
Distribution
Return | Sentiment
$\text{Prob}(r_i > 0 | p_i) = g(p_i)$
Underlying Sentiment
p_i
Purely Positive
Article Word
Distribution: O_+
Purely Negative
Article Word
Distribution: O_-


Realized article d_i
Realized return r_i
Distribution
Return | Sentiment
Prob($r_i > 0 | p_i$) = $g(p_i)$
Purely Positive
Article Word
Distribution: O_+
Purely Negative
Article Word
Distribution: O_-
Mixture
Model
Underlying Sentiment
p_i


Realized article d_i
Realized return r_i
Distribution
Text | Sentiment
$\text{Prob}(d_{i,[S]}|p_i) = MN(s_i, p_i O_+ + [1 - p_i] O_-)$
Distribution
Return | Sentiment
$\text{Prob}(r_i > 0|p_i) = g(p_i)$
Purely Positive Article Word Distribution: O_+
Purely Negative Article Word Distribution: O_-
Mixture Model
Underlying Sentiment p_i


Realized article d_i
$\hat{p}_i$
Realized return r_i
Distribution
Text | Sentiment
$\text{Prob}(d_{i,[S]}|p_i) = MN(s_i, p_i O_+ + [1 - p_i] O_-)$
Distribution
Return | Sentiment
$\text{Prob}(r_i > 0|p_i) = g(p_i)$
Purely Positive Article Word Distribution: O_+
Purely Negative Article Word Distribution: O_-
Mixture Model
Underlying Sentiment p_i

SSESTM Estimator: An Overview

Supervised Sentiment Extraction via Screening and Topic Modeling

1. **Correlation screening** for the most **sentiment-charged** words
   - Massive dimension reduction by screening out **sentiment-neutral** words ($d_i \to d_{i,[\hat{S}]}$)
   - Screening threshold is a tuning parameter
   - Standard ML tool (e.g. Fan and Lv 2008)

2. **Topic model** for estimating sentiment word values
   - Estimate $\hat{p}_i$ as **rank of return** in training data set
   - O_+, O_- estimated from regression of $d_{i,[\hat{S}]}$ on $\hat{p}_i$
   - Words with closer correspondence to positive returns have higher weight in positive "topic"
   - Standard ML tool (e.g. Blei, Ng, and Jordan 2003)

3. **Scoring new articles**
   - Use these estimates and penalized MLE to extract sentiment, p, for each new event **not in** training data
   - Standard ML tool

Step 1: Correlation Screening

- Our screening procedure calculates the frequency with which word j co-occurs with a positive return. This is measured as

$$f_j = \frac{\#\text{ articles including word } j \text{ AND having } \text{sgn}(y) = 1}{\#\text{ articles including word } j}$$

  for each $j = 1, ..., m$.

- Equivalently, f_j is the slope coefficient of a cross-article regression of $\text{sgn}(y_i)$ on a dummy variable for whether word j appears in article i.

- Suppose $(\alpha_+, \alpha_-, \kappa)$ are tuning parameters,

$$\widehat{S} = \left\{ j : f_j \geq 1/2 + \alpha_+, \text{ or } f_j \leq 1/2 - \alpha_- \right\} \cap \{ j : k_j \geq \kappa \},$$

  where k_j is the number of articles including word j.

Step 2: Topic Modeling

- Let $\widetilde{d}_{i,[S]} = d_{i,[S]}/s_i$ denote the vector of word frequencies. The model implies that

$$\mathbb{E}\widetilde{d}_{i,[S]} = \mathbb{E}\frac{d_{i,[S]}}{s_i} = p_i O_+ + (1 - p_i)O_-,$$

  or, in matrix form, writing $O = (O_+, O_-)$,

$$\mathbb{E}\widetilde{D} = OW, \qquad \text{where} \quad W = \begin{bmatrix} p_1 & \cdots & p_n \\ 1 - p_1 & \cdots & 1 - p_n \end{bmatrix}, \quad \text{and} \quad \widetilde{D} = [\widetilde{d}_1, \widetilde{d}_2, \ldots, \widetilde{d}_n].$$

- For each article i in the training sample $i = 1, ..., n$, we set

$$\widehat{p}_i = \frac{\text{rank of } y_i \text{ in } \{y_l\}_{l=1}^n}{n}.$$

Step 3: Scoring New Articles

- Let d be the article's count vector and let s be its total count of sentiment-sensitive words for a new article. Then we have

$$d_{[S]} \sim \text{Multinomial}\Big(s,\ pO_+ + (1-p)O_-\Big),$$

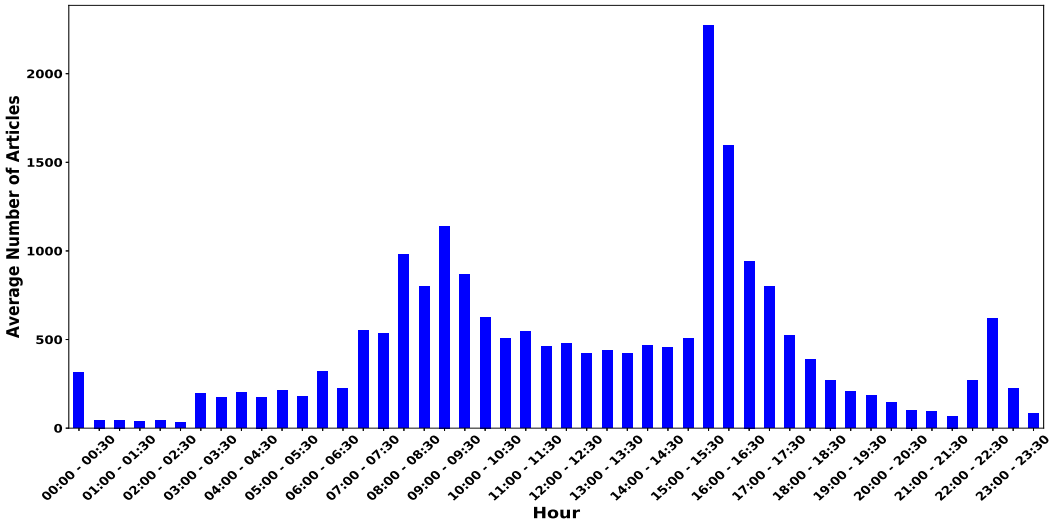
  and given $\widehat{S}$ and $\widehat{O}$, we can estimate p using maximum likelihood estimation (MLE).
- We add a penalty term, $\lambda \log(p(1-p))$, in the likelihood function, which shrinks the estimate towards a neutral sentiment score of $1/2$.
- For most articles, their sentiment is neutral, so imposing such a prior improves the estimates.

Empirical Data: Dow Jones Newswires 1989–2017

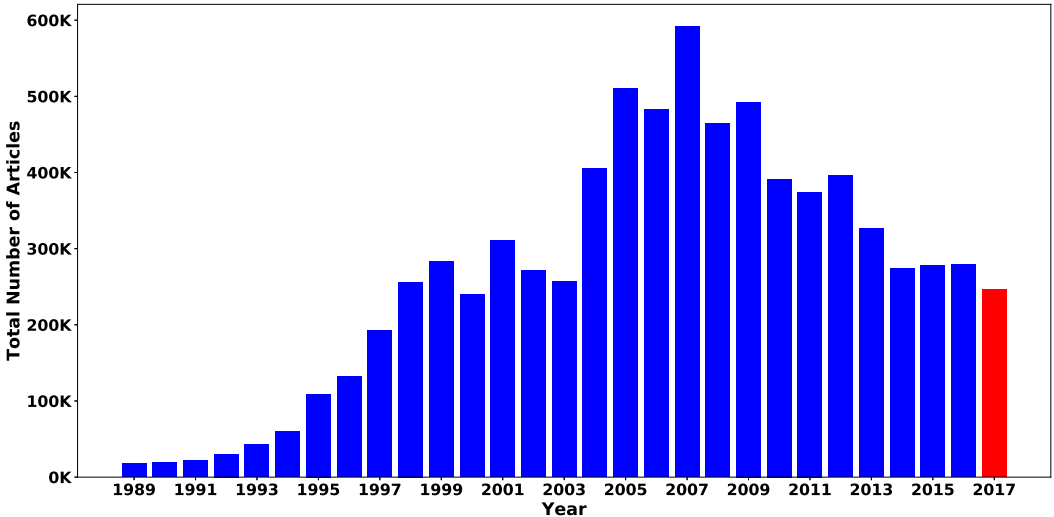
Filter	Remaining Sample Size	Observations Removed
Total Number of Dow Jones Newswire Articles	31,492,473	
Combine chained articles	22,471,222	9,021,251
Remove articles with no stocks tagged	14,044,812	8,426,410
Remove articles with more than one stocks tagged	10,364,189	3,680,623
Number of articles whose tagged stocks have three consecutive daily returns from CRSP between Jan 1989 and Dec 2012	6,540,036	
Number of articles whose tagged stocks have open-to-open returns from CRSP since Feb 2004	6,790,592	
Number of articles whose tagged stocks have high-frequency returns from TAQ since Feb 2004	6,708,077	

- Sources: Press release wire, WSJ, Barron's, MarketWatch + host of Dow Jones realtime services

Average Number of Articles By Half Hour



Annual Time Series of the Total Number of Articles



Data Cleaning

- Normalization, 1) removing proper nouns, and changing all words to lower case letters; 2) expanding contractions such as "haven't" to "have not"; and 3) deleting numbers, punctuations, special symbols, and non-English words
- Stemming and lemmatizing, which group together the different forms of a word to analyze them as a single root word, e.g., "disappointment" to "disappoint", "likes" to "like", and so forth
- Tokenization, which splits each article into a list of words
- Remove stop words such as "and", "the", "is", and "are"

⋆ A limitation of the bag of words approach is that it ignores the context (e.g., negation).

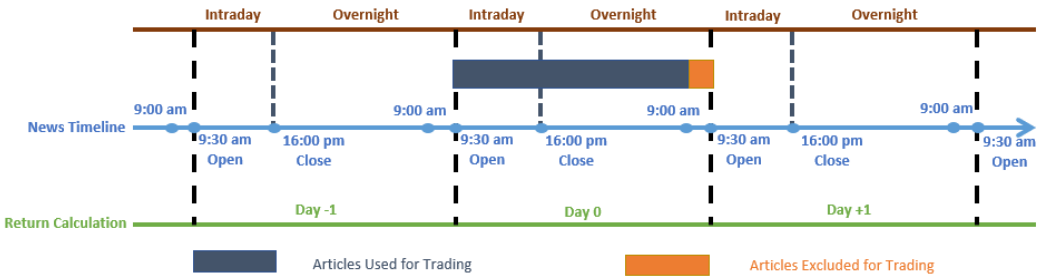
Empirical Strategy

Training (In-sample, 15-year rolling window, 10 year training + 5 year validation)

- Match articles published on day t with return from days $t-1$ through $t+1$

Testing (Out-of-sample, from 2004 to 2017)

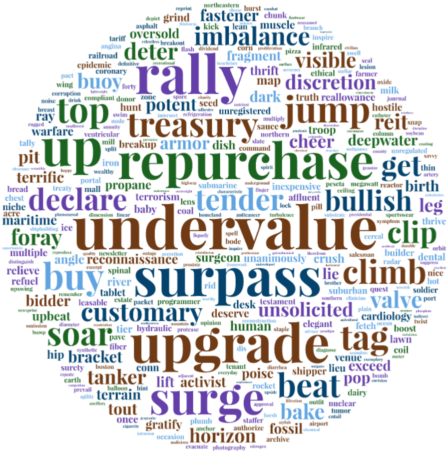
- Using sentiment on day t to predict return on day $t+1$



Sentiment Word Lists

Negative Words

Positive Words

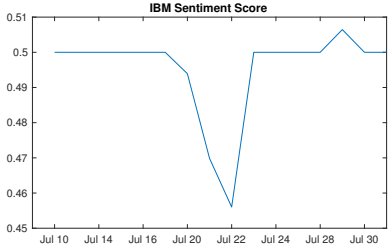


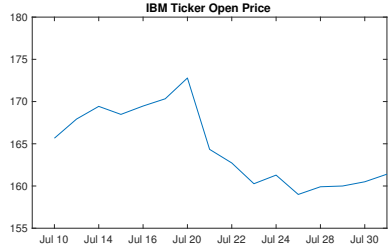
The IBM Example

- The Bag of Words Representation Post Screening:

S	Count
fall	4
erode	1
soften	1
hurt	1
article	1

- The Sentiment Score and IBM stock price:





Forecast Performance Evaluation: A Trading Strategy

Each day, construct out-of-sample estimates of $\hat{p}_i$ for all articles that day

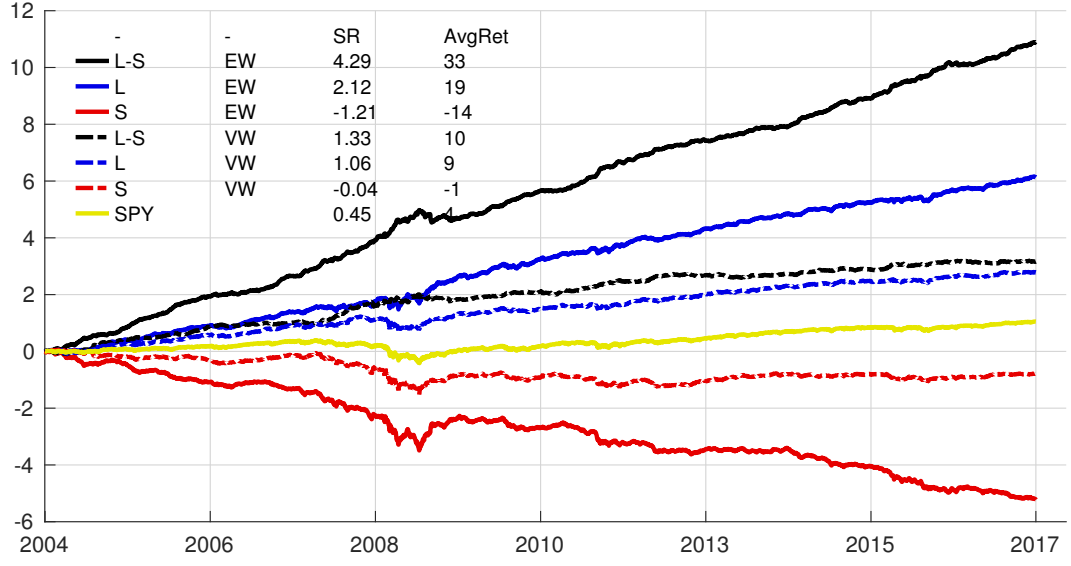
Buy 50 stocks with highest $\hat{p}_i$, sell 50 stocks with lowest

- Equal-weighted and value-weighted constructions
- Zero net investment construction

Evaluate performance from day -10 to day $+10$ relative to article publication date

- Sharpe ratios
- Average daily returns

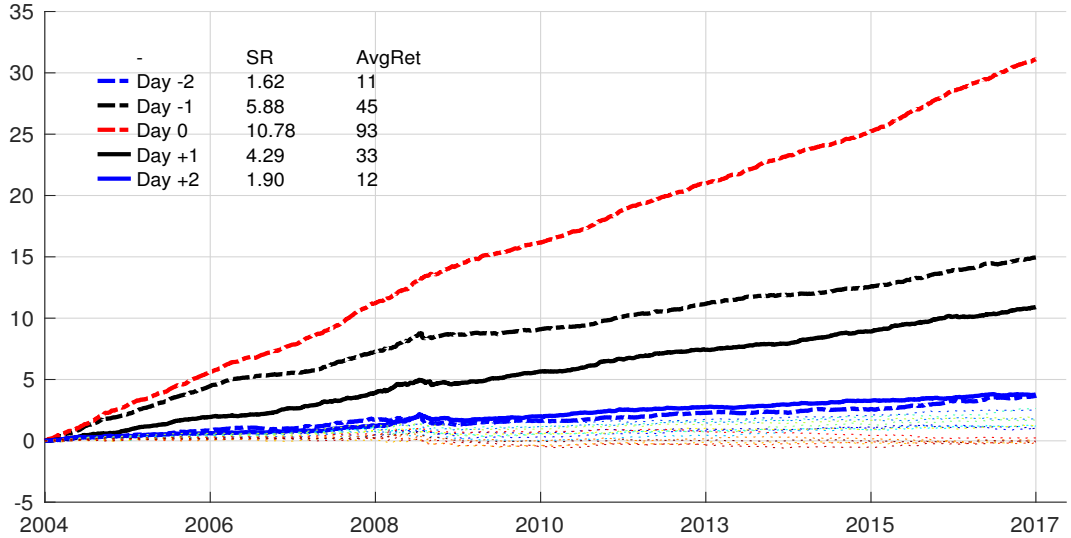
Next-Day Out-of-Sample Trading Strategy Performance



Performance of Daily News Sentiment Portfolios

Formation	Sharpe Ratio	Turnover	Average Return	FF3		FF5		FF5+MOM	
				α	R^2	α	R^2	α	R^2
EW L-S	4.29	94.6%	33	33	1.8%	32	3.0%	32	4.3%
EW L	2.12	95.8%	19	16	40.0%	16	40.3%	17	41.1%
EW S	1.21	93.4%	14	17	33.2%	16	34.2%	16	36.3%
VW L-S	1.33	91.4%	10	10	7.9%	10	9.3%	10	10.0%
VW L	1.06	93.2%	9	7	30.7%	7	30.8%	7	30.8%
VW S	0.04	89.7%	1	4	31.8%	3	32.4%	3	32.9%

Portfolio Performance Day -10,...,-1,0,1,...,10



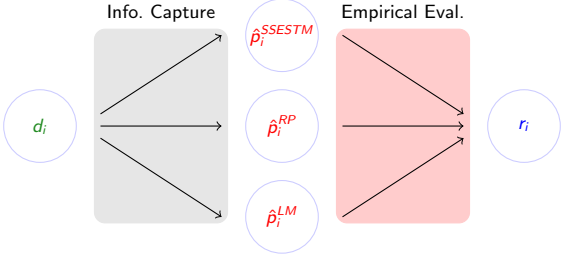
Comparison with RavenPack and Dictionary Methods

Leading vendor of news-based sentiment scores is RavenPack

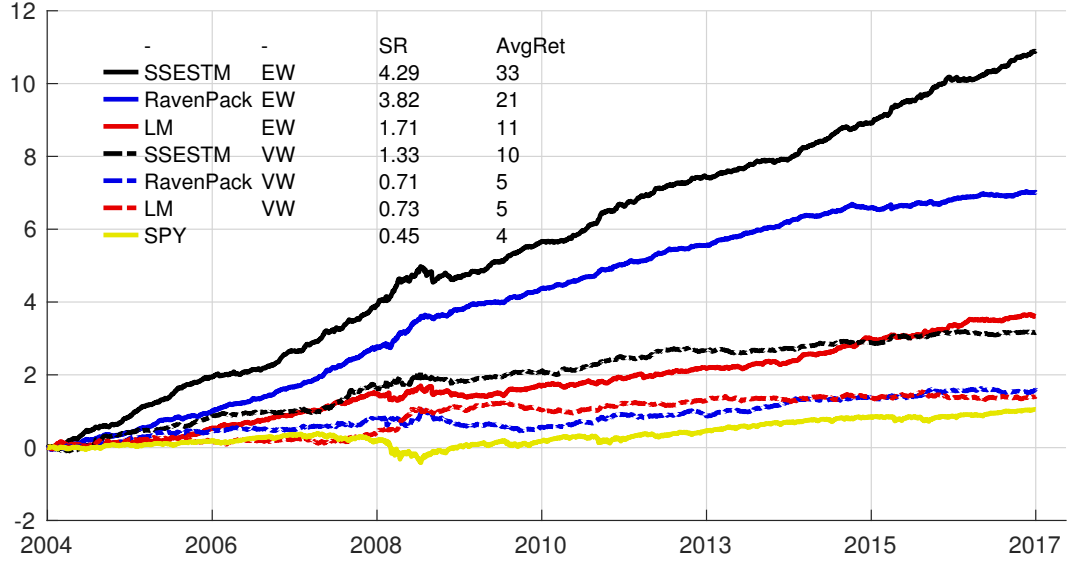
- Widely used by major asset managers
- We use the article-level estimates of p_i that they sell

Another benchmark is dictionary-based sentiment scoring

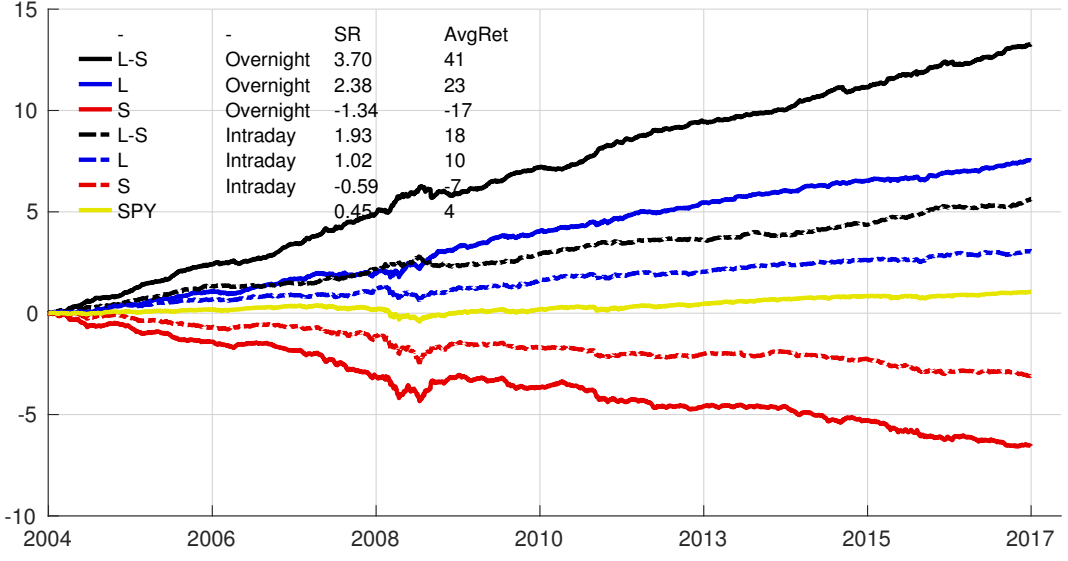
- We consider Loughran-McDonald finance sentiment dictionary
- Calculate LM-based estimate of p_i for each article



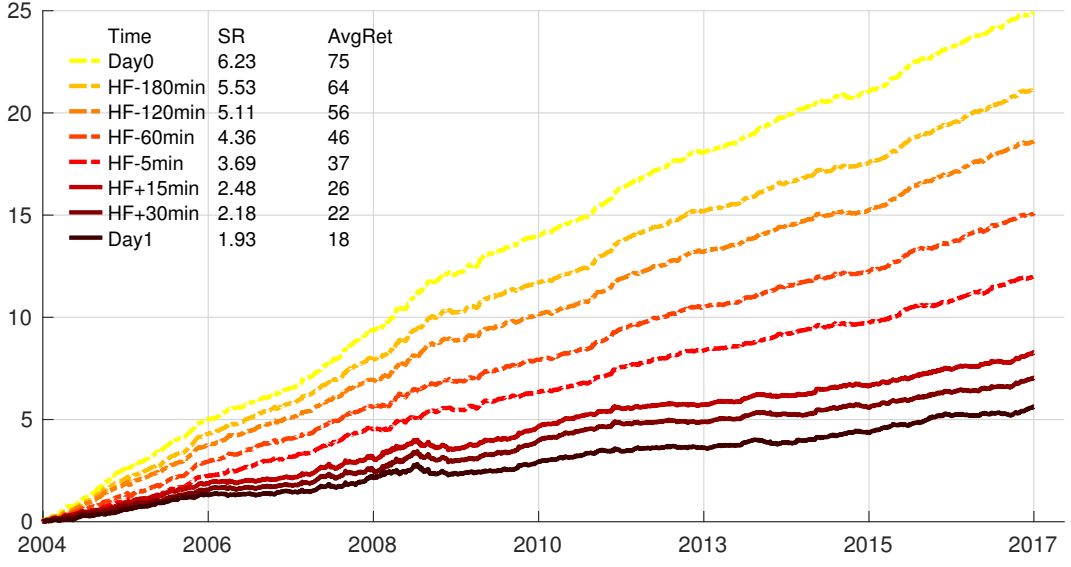
Comparison with RavenPack and Dictionary Methods



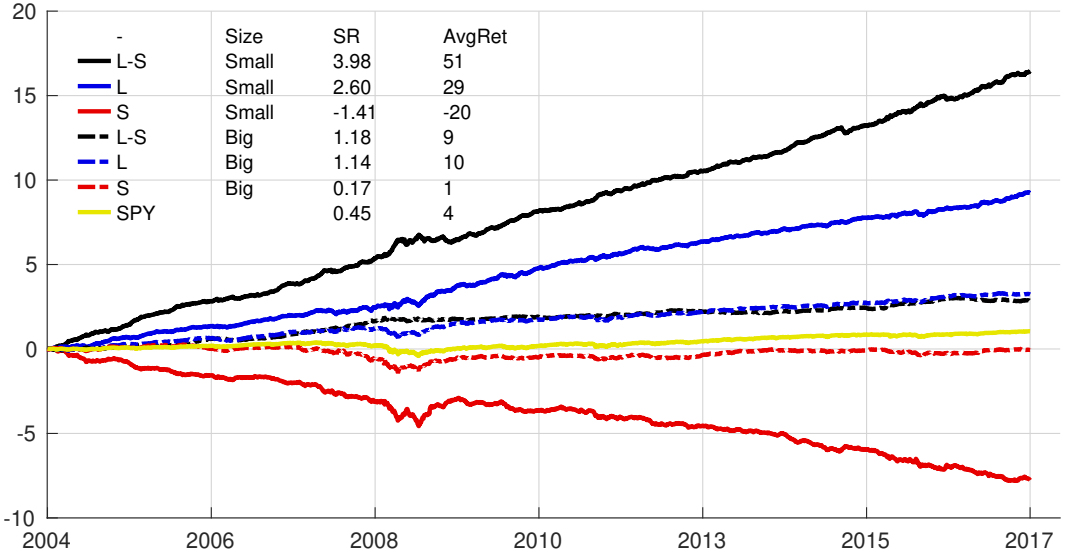
Intraday vs. Overnight



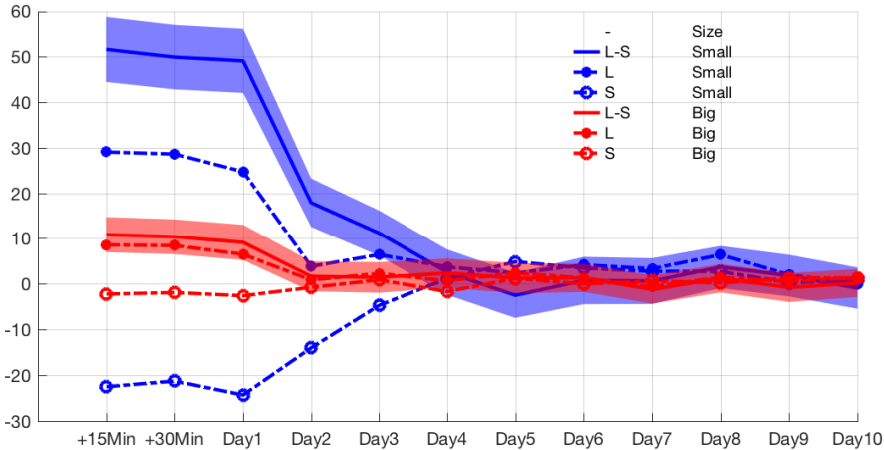
High Frequency Trading



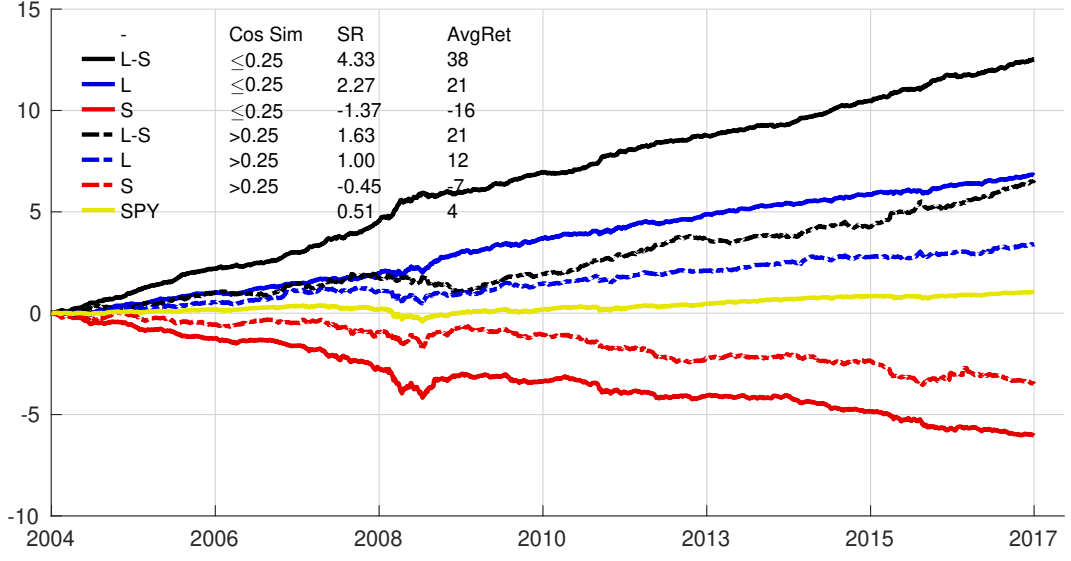
Big vs Small Size



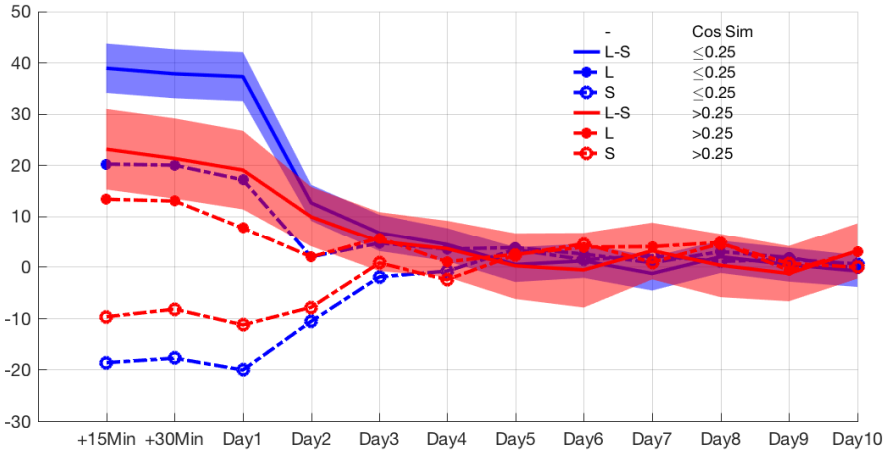
Speed of News Assimilation (Big vs Small)



Fresh vs Stale News



Speed of News Assimilation (Fresh Versus Stale News)



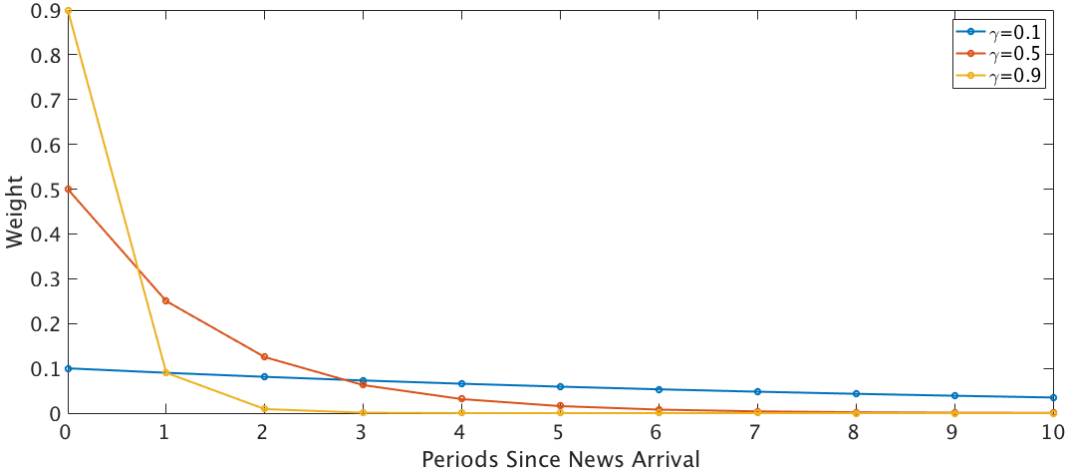
Transaction Costs

1. We assume that each portfolio incurs a daily transaction cost of **10bps**
Frazzini, Israel, and Moskowitz (2018)

2. We propose a novel trading strategy **exponentially-weighted calendar time portfolio** that directly reduces portfolio turnover and hence trading costs

- On each day t, we liquidate a fixed proportion γ of all existing positions, and reallocate that γ proportion to an equal-weighted portfolio based on day t news.
- The turnover parameter simultaneously governs both the size of the weight spike at news arrival (the amount of portfolio reallocation) as well as the exponential decay rate for existing weights.

EWCT Weights



Transaction Costs

		Gross		Net	
γ	Turnover	Return	Sharpe Ratio	Return	Sharpe Ratio
0.1	0.08	5.18	1.77	3.58	1.17
0.2	0.17	9.74	2.93	6.31	1.84
0.3	0.27	13.71	3.61	8.37	2.16
0.4	0.36	17.24	4.03	9.98	2.28
0.5	0.46	20.43	4.26	11.23	2.30
0.6	0.56	23.32	4.38	12.17	2.25
0.7	0.66	25.97	4.43	12.88	2.15
0.8	0.75	28.43	4.42	13.39	2.04
0.9	0.85	30.74	4.37	13.74	1.92

Note: The table reports the performance of equally-weighted long-short EWCT portfolios based on SESTM scores. The EWCT parameter is γ. Average returns are reported in basis points per day and Sharpe ratios are annualized. Portfolio average daily turnover is calculated as $\frac{1}{T}\sum_{t=1}^{T}\left(\sum_i \left| w_{i,t+1} - \frac{w_{i,t}(1+y_{i,t+1})}{\sum_j w_{j,t}(1+y_{j,t+1})} \right|\right)$.

Theory

1. Sure Screening: As $n, m \to \infty$, with probability $1 - o(1)$,

$$|f_j - 1/2| \begin{cases} \geq 2\theta \frac{|O_{+,j} - O_{-,j}|}{O_{+,j} + O_{-,j}} + \frac{C\sqrt{\log(m)}}{\sqrt{n \min\{1, \bar{s}(O_{+,j} + O_{-,j})\}}}, & \text{for } j \in S, \\ \leq \frac{C\sqrt{\log(m)}}{\sqrt{n \min\{1, \bar{\Omega}_{\cdot,j}\}}}, & \text{for } j \in N. \end{cases}$$

where

$$\theta \equiv \frac{\sum_{i=1}^n s_i \left(p_i - \frac{1}{2}\right)\left[g(p_i) - \frac{1}{2}\right]}{\sum_{i=1}^n s_i}, \quad \Omega_i = \mathbb{E} d_{i,[N]}, \quad \bar{\Omega}_{\cdot,j} = \frac{1}{n}\sum_{i=1}^n \Omega_{i,j}.$$

Theory (cont'd)

Therefore, as long as

$$n\theta^2 \underbrace{\min_{j\in S} \frac{(O_{+,j}-O_{-,j})^2}{(O_{+,j}+O_{-,j})^2}}_{\Delta^*} \geq \frac{\log^2(m)}{\min\{1, \underbrace{\bar{s}\min_{j\in S}(O_{+,j}+O_{-,j})}_{\ell_S}, \underbrace{\min_{j\in N}\bar{\Omega}_{\cdot,j}}_{\ell_N}\}},$$

we have

$$\mathbb{P}(\widehat{S}=S) = 1-o(1).$$

- n: number of training articles
- m: size of dictionary
- θ: sensitivity of returns to sentiment (related to g')
- Δ^*: separability between O_+ and O_-
- ℓ_S, ℓ_N: quantities related to per-article counts of sentiment-charged and sentiment-neutral words

Theory (cont'd)

2. Consistency: reorganizing (O_+, O_-) into a *vector of frequency*, F, and a *vector of tone*, T:

$$F = \frac{1}{2}(O_+ + O_-), \qquad T = \frac{1}{2}(O_+ - O_-),$$

we have

$$\|\widehat{F} - F\|_1 \leq C\sqrt{\frac{|S|\log(m)}{n\bar{s}}}, \qquad \|\widehat{T} - \rho T\|_1 \leq C\sqrt{\frac{|S|\log(m)}{n\bar{s}}}.$$

where

$$\rho = \frac{12}{n}\sum_{i=1}^{n}\Big(p_i - \frac{1}{2}\Big)\Big(\widehat{p}_i - \frac{1}{2}\Big).$$

Theory (cont'd)

3. Scoring Error on New Article: Given a new article with sentiment p, define the *rescaled sentiment* as

$$p^* = \frac{1}{2} + \rho^{-1}\Big(p - \frac{1}{2}\Big).$$

Then we have, with probability approaching 1

$$|\widehat{p} - p^*| \leq C \min\{err_n, |p^* - \frac{1}{2}|\},$$

where

$$err_n = \frac{1}{\rho\sqrt{\Theta}}\left(\frac{\sqrt{|S|\log(m)}}{\rho\sqrt{n\bar{s}\Theta}} + \frac{1}{\sqrt{s}}\right), \qquad \text{where} \quad \Theta = \sum_{j\in S}\frac{(O_{+,j} - O_{-,j})^2}{O_{+,j} + O_{-,j}}.$$

4. Let $SR(\hat{p}, p)$ be the Spearman's rank correlation between $\{\hat{p}\}_{i=1}^N$ and $\{p_i\}_{i=1}^N$. As $n, m, N \to \infty$,

$$\mathbb{E}[SR(\hat{p}, p)] \to 1.$$

Simulations

- We assume the data generating process of the positive, negative, and neutral words in each article follow:

$$d_{i,[S]} \sim \text{Multinomial}\Big(s_i,\ p_i O_+ + (1-p_i)O_-\Big), \quad d_{i,[N]} \sim \text{Multinomial}\Big(n_i,\ O_0\Big),$$

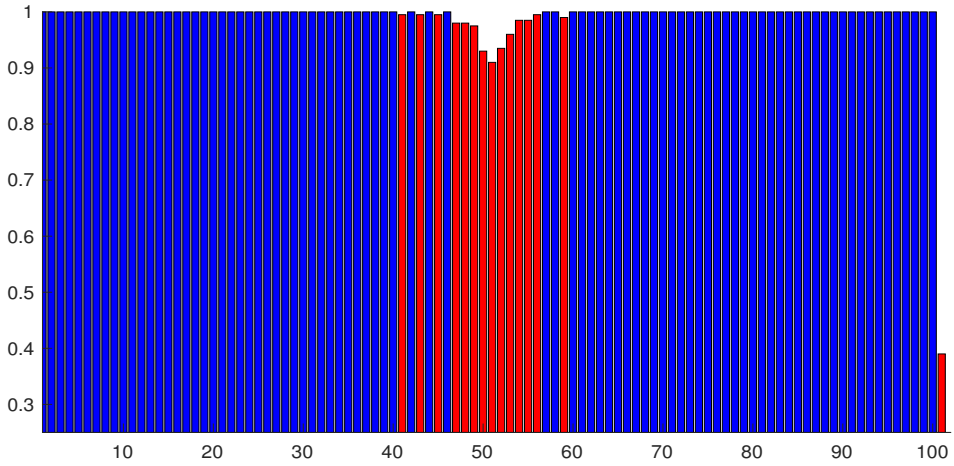
where $p_i \sim \text{Unif}(0,1)$, $s_i \sim \text{Unif}(0, 2\bar{s})$, $n_i \sim \text{Unif}(0, 2\bar{n})$, and for $j = 1, 2, \ldots, S$,

$$O_{+,j} = \frac{2}{|S|}\left(1 - \frac{j}{|S|}\right)^2 + \frac{2}{3|S|} \times 1_{\{j < \frac{|S|}{2}\}}, \quad O_{-,j} = \frac{2}{|S|}\left(\frac{j}{|S|}\right)^2 + \frac{2}{3|S|} \times 1_{\{j \geq \frac{|S|}{2}\}},$$

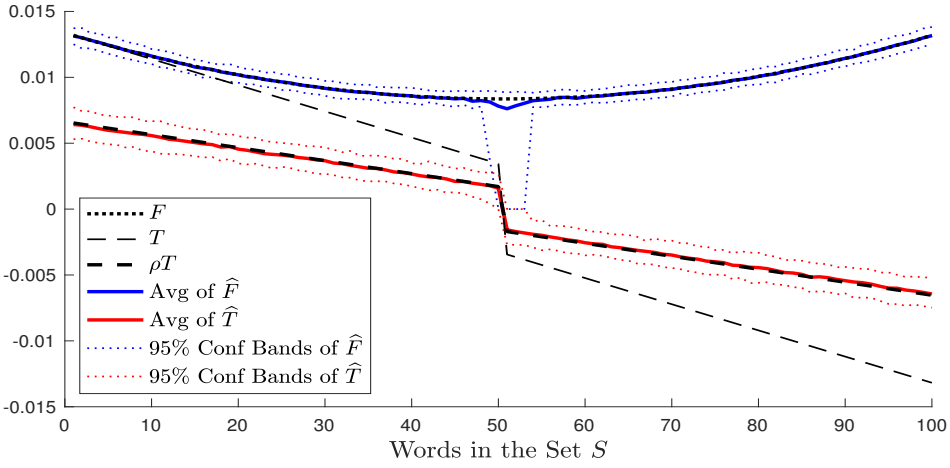
and $O_{0,j} \sim \frac{1}{m-|S|}\text{Unif}(0,2)$, for $j = |S|+1, \ldots, m$. As a result, the first $|S|/2$ words are positive, the next $|S|/2$ words are negative, and the remaining ones are neutral with frequencies randomly drawn from a uniform distribution.

- The sign of returns follows a logistic regression model: $\mathbb{P}(y_i > 0) = p_i$, and its magnitude $|y_i|$ follows a Student t-distribution with the degree of freedom parameter set at 4.

Screening Results in Simulations



Estimation Results in Simulations



Prediction Results in Simulations

	benchmark	$\bar{s} \downarrow$	$n \downarrow$	$m \downarrow$	$\lvert S \rvert \downarrow$
Avg S-Corr	0.850	0.776	0.834	0.857	0.852
Std Dev	0.0014	0.0043	0.0024	0.0025	0.0009

Note: In this table, we report the mean and standard deviation of Spearman's correlation estimates across Monte Carlo repetitions for a variety of cases. The parameters in the benchmark case are set as: $|S| = 100$, $m = 500$, $n = 10,000$, and $\bar{s} = 10$. In each of the remaining columns, the corresponding parameter is decreased by half, whereas the rest three parameters are fixed the same as the benchmark case.

Conclusion

Introduce new text-mining model for extracting sentiment information from text

- Supervised: Customized to research context at hand
- High-dimensional: Generative ML framework manages/exploits complexity of text

Develop estimation approach and statistical guarantees

- Recovers "true" sentiment ranks in large samples with minimal guarantees
- SSESTM is easy to use and it's a "white" box!

Empirical evaluation through portfolio choice

- SSESTM excels at extracting return-predictive signals from *Dow Jones Newswires*
- Outperforms the industry standard RavenPack