A supervised learning framework for textual sentiment analysis conditioned to forecasting targets

Pedro Freitas, Kristin Lomicka, Alexandros Pappas

Abstract: We implemented and explored the novel methodology for sentiment analysis introduced by Ke et al. in their 2019 article titled "Predicting Returns with Text Data" to understand the conditions required to replicate the results. The authors present a supervised topic model called "SESTM" (Sentiment Extraction via Screening and Topic Modeling) that uses correlation screening to create sentiment scores customized for individual research applications, which in this case is the prediction of financial returns using Dow Jones Newswires financial news articles. Substantiating the findings in the paper, the sentiment scores produced by SESTM are strong predictors of price responses to new information and performed well when compared to popular benchmarks. We validated this conclusion by analyzing the performance of a systematic trading strategy using the sentiment scores derived from the model.

Keywords: supervised learning, topic model, sentiment analysis, stock returns, SESTM, sentometrics, systematic trading

1. Introduction

The use of alternative data like text, sound, and image in machine learning has been getting more popular following advances in the field of natural language process, in computing power and in software capabilities. The financial market is no different and participants have been experimenting with applying those techniques to gain an extra edge in an increasingly competitive industry.

Text data is the most used alternative data in the financial ecosystem and usually comes in the form of company reports, central bank statements, financial news, and social media posts, serving as an important source of information for investors. Sentiment analysis refers to the use of natural language processing (NLP) and machine learning techniques to systematically identify, extract, and quantify a text in a range spanning from positive to negative. Its applications in the financial market have been the subject of numerous papers with the end goal of exploiting the relationship between the text and price, returns, volatility or another statistic to improve portfolio performance.

As discussed in Gentzkow et al. (2019)[7], the most signficant way text data differs from other data is the dimensionality. As such, dictionary-based methods are currently the most commonly used techniques for sentiment analysis of financial texts. These methods compare the frequency of

terms in a document with those in a pre-defined for-purpose lexicon. Dictionary-based methods work well when prior information is strong and the availability of appropriately labeled training data is limited. Modern methods leveraging statistical tools commonly employed for high dimensional data, however, are expected to outperform dictionary-based methods in a substantial share of cases.

One of these modern approaches is word embeddings, which involves creating richer representations of words in a metric space that takes into consideration similarities between words or sentence syntax as opposed to only term frequencies. Approaches such as this that exploit linguistic principles around grammatical structures and interactions within text are starting to play an important role in the next generation of analysis using text-as-data. [7]

The objective of our paper is to explore a novel methodology for sentiment analysis introduced by Ke et al. in their 2019 article titled "Predicting Returns with Text Data" [9] and understand the conditions required to replicate the results. The authors present a supervised topic model called "SESTM" (Sentiment Extraction via Screening and Topic Modeling) that uses correlation screening to create sentiment scores customized for individual research applications, which in this case is the prediction of financial returns using Dow Jones Newswires financial news articles. As opposed to a lexicon-based approach to address the high dimensionality inherent in text data, SESTM uses correlation screening to identify the correlation between words in the corpus and the target variable, resulting in the identification of the most relevant sentiment-charged words for the purpose.

Ke et al. (2019) claim that the SESTM model: (1) creates customized sentiment scores for individual research applications using basic statistical tools, (2) is clearly interpretable ("white box"), (3) is suitable to analyze big data with low computational cost and (4) produces results that outperform standard NLP models for the specific task of predicting returns. In order to explore each of these claims, we implemented the SESTM model using two datasets of vastly different sizes and assessed the model performance by examining the accuracy and Spearman Rank Correlation of the estimated sentiment scores. In addition, we measured the predictive ability of the sentiment scores on returns by backtesting a simple porfolio strategy. Lastly, we benchmarked the SESTM model results by comparing them to two popular NLP techniques for sentiment analysis, a lexicon-based model utilizing the Sentometrics package in R [2] and a word-embedding approach leveraging Word2Vec embeddings and an XGBoost regression algorithm.

The rest of the paper is organized as follows: Section 2 presents a brief literature review. In Section 3, we introduce the main theoretical components of the SESTM model. Section 4 presents our empirical analysis, including a description of data and pre-processing steps, the approach for training the SESTM model and scoring new articles, and an analysis of the results. The lexicon-based approach and Word2Vec word embedding model used to benchmark the SESTM results are described in Section 5. Section 6 briefly discusses the systematic trading strategy derived to evaluate the out-of-sample prediction power of the model. Lastly, Section 7 concludes our findings and proposes opportunities for future exploration.

2. Literature Review

At least since 1933, with the work by Alfred Cowles [5], the impact of news on the stock market is the subject of research. At that time, Cowles manually read the newspaper and classified the Wall Street Journal as bullish, bearish, or doubtful. A market-timing strategy based on his classification Wall Street Journal editorials, however, underperformed the Dow Jones Industrial Average by 3.5 percentage points per year.

The work by Antweiler and Frank (2004) analyzed internet stock message boards to predict stock returns. They manually classified a training data set of 1,000 messages on stocks internet boards. Based on this training data set, the entire sample (1,559,621 messages) is then classified to obtain buy, hold, or sell signals using a Naive Bayes approach. The signals are aggregated into indices that measure the bullishness of each stock message board during each period. Their findings include that stock messages boards help predict market volatility and their effect on stock returns is statistically significant but economically small [1].

More recently, dictionary-based methods gained strength with Tetlock (2007) [14] analyzing media sentiment and the stock market. His research involved scoring the Wall Street Journal's "Abreast of the Market" column into a vector of sentiment in seventy-seven different sentiment dimensions based on the Harvard IV-4 psychosocial dictionary. The time series of daily sentiment scores for each category were consolidated into a single principal component, named the "pessimism factor" due to its close association with the "pessimism" dimension. The pessimism score was shown to be negatively associated with one-day-ahead returns on the Dow Jones Industrial Average.

In 2011, Loughran and Mcdonald [10] showed that a finance-specific dictionary outperforms general-purpose dictionaries for financial applications. Their initiative came from the understanding that most of the negative word counts according to the Harvard list are attributable to words that are typically not negative in a financial context. In their own words, "Words such as tax, cost, capital, board, liability, foreign, and vice are on the Harvard list. These words also appear with great frequency in the vast majority of 10-Ks, yet often do no more than name a board of directors or a company's vice-presidents. Other words on the Harvard list, such as mine, cancer, crude (oil), tire, or capital, are more likely to identify a specific industry segment than reveal a negative financial event."

Bollen, Mao, and Zeng (2011) [4] investigated whether public sentiment, as expressed in large-scale collections of daily Twitter posts, could be used to predict the stock market. Public sentiment was measured across 7 different dimensions (Positiveness, Calm, Alert, Sure, Vital, Kind and Happy), and daily time series of those was built using dictionary-based methods. The results showed a significant predictive correlation between Twitter messages and the stock market.

In Manela and Moreira (2017)[11], they applied support vector machines for predicting turbulence in financial markets based on news articles. They dealt with high dimensionality using a penalized least squares objective to identify a small subset of words whose frequencies are most

useful for the task. They constructed an index of news-implied market volatility based on text from the Wall Street Journal from 1890–2009 and found that the levels of news-implied volatility predict future stock market returns at frequencies from six months up to twenty four months.

3. SESTM Model

As mentioned in the introduction, the SESTM model is a simple, interpretable, correlation-based sentiment analysis model that is specifically adapted to the problem of return prediction. Rather than using a pre-defined sentiment dictionary such as Harvard IV or LM, SESTM identifies the most positively and negatively sentiment charged words conditioned on the stock returns as a proxy for sentiment. The model consists of three parts, which are described in more detail below. The first step identifies the most relevant words, or features, for indicating sentiment from a large dictionary of words via correlation screening. The second step weights each of the terms based on their importance for predicting sentiment by means of a two-topic supervised topic model. Lastly, the third step assigns a sentiment score to each article, based on the topic model.

3.1. Notation

The input into the model is a corpus of news articles. Each article is tagged with a single stock, which has an associated 3-day (t-1 to t+1) return. In order to establish notation, the following terms are use:

- n: The number of news articles
- m: The length of the dictionary
- $D = [d_1, ..., d_n]$: An nxm document-term matrix
- S: The set of sentiment-charged words
- $D_{\cdot,[S]}$: The sub-matrix of D corresponding to the words in the sentiment-charged word list S
- $d_{i,[S]}$: The vector of sentiment-charged word counts for document i, corresponding to the i^{th} row of $D_{\cdot,[S]}$
- y_i: The return associated with the stock tagged to article i
- $p_i \in [0,1]$: The sentiment score for article i
- f_j: The frequency with which word j co-occurs with a positive return

3.2. Step 1: Create Sentiment Charged Word List (Feature Selection)

In this first step, we isolate both the positive and negatively charged words, collectively referred to as sentiment-charged words, in the dictionary from those with neutral sentiment. The purpose of this is to reduce the dimensions of the feature space so that the topic model can be estimated on the lower-dimensional space. The stock returns are used as a proxy for document sentiment based on the intuition that if a word frequently co-occurs in documents associated with positive (or negative) returns, that word must be indicative of positive (or negative) sentiment. As a first step, the formula below is used to estimate the frequency with which word j co-occurs with a positive return for each j = 1,...,m.

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

f_j can be interpreted such that words with a higher f_j appear more times in articles associated with positive returns, and thus have positive sentiment. Those with a lower f_j appear fewer times and are less positively charged, or in fact, negatively charged. Words occurring equally in documents associated with positive and negative returns will have an f_j of 0.5. Once f_j is determined, the sentiment charged words are selected by choosing those with an $f_j > 0.5+\alpha_+$ for positive sentiment terms and $f_j > 0.5-\alpha_+$ for negative sentiment terms, where $\alpha_{+/-}$ serves as a threshold for filtering the sentiment charged words from those with neutral sentiment. The list of words satisfying these requirements is the estimate for the set of sentiment-charged words, S:

$$\hat{S} = \{j : f_j \geq 0.5 + \alpha_+, \text{ or } f_j \leq 0.5 - \alpha_-\} \cap \{j : k_j \geq \kappa\}$$

This step leverages an approach called sure screening, which is a means of dimension reduction via correlation learning used to filter out features that are weakly correlated with the response variable. It was first introduced by Fan and Lv (2008) [6] as a means to address the concerns in performance of the Dantzig selector using L_1 regularization in ultra high dimensions. Sure screening can reduce the dimension of the feature space to one below that of the sample size, in turn improving both the speed and accuracy of variable selection. This approach is said to have the "sure screening property" if all the important variables survive after applying a variable screening procedure with probability tending to 1. Ke et al. (2019) assert that SESTM has the "sure screening property" as $|f_j - 0.5|$ is large for sentiment-charged words and small for sentiment neutral words, indicating that the screening step is meaningful.

3.3. Step 2: Weight Sentiment Charged Words Based on Relevance (Topic Model)

The objective of this step is to create a two-topic topic model using a supervised learning technique. The outputs are the probability distributions over each word for the positive and negative topics, represented as $O = [O_+, O_-]$. As mentioned above, the stock returns y are used as a proxy for document sentiment. The topic model assumes that $d_{i,[S]}$, the sentiment-charged word counts, are generated by a the mixture multinomial distribution below where s_i, the total count of sentiment-charged words in document i, determines the scale of the distribution.

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

Given this distribution, the expected value of the count of the sentiment charged words in document i is:

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

The expectation can be re-expressed in matrix form as follows:

$$E[\tilde{D}'] = OW, \text{ where } W = \begin{bmatrix} p_1 & \dots & p_n \\ 1 - p_1 & \dots & 1 - p_n \end{bmatrix}, \text{ and } \tilde{D} = [\tilde{d}_1, \tilde{d}_2, ..., \tilde{d}_n]'$$

This means that in the model, O can be estimated by,

$$\hat{O} = \hat{D}\hat{W}'(\hat{W}\hat{W}')^{-1}, \text{ where } \hat{W} = \begin{bmatrix} \hat{p}_1 & \dots & \hat{p}_n \\ 1 - \hat{p}_1 & \dots & 1 - \hat{p}_n \end{bmatrix}$$

The O matrix has some interesting properties as it incorporates both the frequency and tone of each word. The frequency is represented by $F = O_+ + O_-$ and $T = O_+ - O_-$ represents the tone.

The final piece is $\hat{p}_i$, the estimated sentiment score for each article, which is required to estimate $\hat{W}$. Given the assumption that the returns y are a proxy for article sentiment, the sentiment score can by approximated by taking the standardized rank of the returns for all articles in the training sample.

$$\hat{p}_i = \frac{\text{rank of } y_i \text{ in } \{y_i\}_{i=1}^n}{n}$$

As noted in Ke et al. (2019), "the important feature of this model is that, for a given event i, the distribution of sentiment-charged word counts and the distribution of returns are linked through the common parameter, p_i. Returns supervise the estimation and help identify which words are assigned to the positive versus negative topic."

It is imperative to point out the importance of the estimated document sentiment, $\hat{p}_i$, and its rank in this model. The statistical efficiency gain of supervised learning in the short article setting was central to the authors' decision to use a supervised, rather than unsupervised approach. They found that the SESTM model achieved the best possible error rate of unsupervised methods, but

with a smaller document length to dictionary size than than is required by unsupervised models.[8] The author's found that one of the downsides of a supervised approach is that the predicted tone, T, is discounted by a factor of ρ. As a result, there may exist an error between p_i and $\hat{p_i}$, but the rank remains the same. Accordingly, when incorporating the model results into a trading portfolio, one must consider the rank of $\hat{p_i}$ rather than its distinct value.

3.4. Step 3: Score New Articles

Ke et al. (2019) use a penalized maximum likelihood estimation approach for scoring new articles (i.e., estimating their sentiment score p_i), selected for its statistical efficiency. Within the maximum likelihood estimation framework, the model maximizes the probability of observing the mix of sentiment charged words in each article i given the article's sentiment score p_i, $P(d_{i,[S]}|p_i)$. The parameters of the model are the $\hat{S}$ and $\hat{O}$ estimated in steps 1 and 2. Recall that the document-term vector for sentiment charged words, $d_{i,[S]}$, follows the multinomial distribution below.

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

The likelihood function is:

$$\hat{p} = \arg\max \left[\hat{s}^{-1} \sum_{j=1}^{\hat{s}} d_j \log\left(p\hat{O}_{+,j} + (1-p)\hat{O}_{-,j}\right) + \lambda \log(p(1-p)) \right]$$

We then take the derivative of the function,

$$\frac{\delta f}{\delta p} = d_j \left[\frac{1}{p\hat{O}_{+,j} + (1-p)\hat{O}_{-,j}} (\hat{O}_{+,j} - \hat{O}_{-,j}) \right] + \lambda \frac{1}{p} - \lambda \frac{1}{1-p}$$

and use gradient descent to find the optimal value of $\hat{p}$ for each new article.

Note that a penalty term of $\lambda\frac{1}{p} - \lambda\frac{1}{1-p}$ was added to the likelihood function, with a tuning parameter of λ. The addition of the penalty shrinks the score closer a neutral sentiment of 0.5 in an effort to compensate for the limited number of observations and the low signal to noise ratio in the data. The overall effect is similar to imposing a Beta distribution on the sentiment score, leveraging the assumption that most articles have a neutral sentiment.

4. Empirical Analysis

Our corpus consists of news articles sourced from Dow Jones Newswires, a news aggregator that consolidates economic, financial and political news from around the world and makes it available in real-time. Dow Jones Newswires are highly leveraged by financial professionals to stay abreast of fast-moving markets and are a common source of alternative data for portfolio managers. Ke et

al. (2019) benefited from access to an extremely large corpus of more than 10 million[1] Dow Jones Newswires articles from January 1, 1989 to July 31, 2017. After pre-processing, there were a total of 6,540,036 articles in the corpus. After dividing the model into training, validation and test sets of ten years, five years and one year respectively, they were able to train, test and validate the model fourteen times using rolling window estimation. Given that not all investment managers may have access to a corpus of that size, we explored data sets of different sizes in order to understand the impact of training set size on the model results.

4.1. Data and Pre-processing

We performed our analysis on two corpuses of different sizes. The first, termed the small dataset, consisted of 148,167[2] news articles for a selection of 69 stocks from January 1, 2012 to April 8, 2020. The second corpus, termed the large dataset, contained 3.8 million articles, inclusive of all news articles published between January 1, 2016 and May 5, 2020. The two datasets allowed us to experiment with the size of the training set and draw conclusions about the relationship between corpus size and the model results.

Table 1: Data Summary Statistics

	Large Dataset		**Small Dataset**	
	Sample Size	Obs. Removed	Sample Size	Obs. Removed
Total no. of articles	2,883,935	-	148,167	-
No. tagged to a single stock	2,879,943	3992	148,102	65
No. with 20 or more words	1,372,452	1,507,491	95,917	52,185
Maximum article length	1,170	-	1,043	-
No. unique stocks (post-filters)	7,324	-	69	-

In line with the preprocessing performed by Kelly et al. (2019), we removed special characters and punctuation, remove proper nouns, converted the corpus to lowercase, removed stop words, lemmatized and then removed uncommon words. Lastly, we limited the corpus to articles with twenty or more words after pre-processing in order to eliminate noise.[3] After pre-processing, the small dataset contained 95,917 articles associated with 69 stocks and the large dataset contained 1,372,452 aligned to 7,342 stocks.

[1] After combining chained articles and filtering for article with no stock tagged and more than one stock tagged.

[2] After filtering out articles with no stock tagged, multiple stocks tagged or stocks listed on a foreign exchange

[3] We used the NLTK package in Python to remove proper nouns and English stop words, as well as perform the lemmatization.

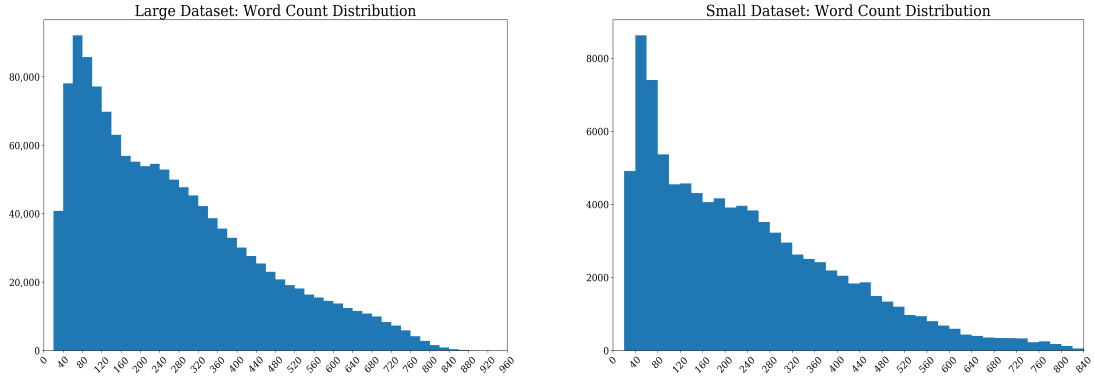


Fig. 1: Word count distributions for the small and large datasets

4.2. Training the SESTM Model and Scoring New Articles

In order to train, tune and test the model, we split our sample into multiple training, validation and test sets and used a rolling window analysis to evaluate the model results over multiple test sets. A rolling window analysis is often used instead of cross validation for time series data as it respects the temporal order of observations. After training the model, the validation set was used to tune the model parameters described below through grid search cross validation. The optimal combination of parameters, those that minimized the loss, were used to score the articles in the test set. The loss was calculated by taking the ℓ^1 norm of the differences between the estimated article sentiment scores and the corresponding standardized return ranks for all observations in the validation set. The tuning parameters are as follows:

1. κ: The bottom limit for the number of articles in which a word must appear in order to be included in the sentiment charged word list. We considered five options based on the (86%, 88%, 90%, 92% and 94%) quantiles of the count distribution of articles in the training set.
2. α: The number of words in each of the positive and negative word lists. We considered three possibilities of (25, 50 and 100), resulting in sentiment charged word lists of length (50, 100 and 200)
3. λ: The model discount factor. A higher number reflects a larger discount on the predicted sentiment score for a new article, bringing the sentiment closer to neutral. Three choices of (1, 5 and 10) were considered.

The small dataset was the original dataset provided to us and it was run with the following train - validate - score split.

1. Model 1: Train small dataset on one year, validate on 60 days, score 30 days of new articles.

2. Model 2: Train small dataset on five years, validate on two years, score on one year of new articles.

However, as will be discussed in the Model Results section, the results with the small dataset were poor. We obtained the large data set and ran it with the following train - validate - test splits in order to see if a larger dataset with a greater variety of stocks would improve the results.[4]

1. Model 3: Train large dataset on one year, validate on 180 days and test on 180 days.
2. Model 4: Train large dataset on 180 days, validate on 180 days and test on 180 days.

4.3. Model Results

A key finding from our research was that the quantity of data and diversity of stocks represented in the corpus is essential to improving the quality of the model results. Model 3, from the large dataset clearly outperformed the others across all measures. This model had the largest average training sample size of 314,758 articles spanning one year, aligned to an average of 5,381 stocks.

Table 2: SESTM Model Results: Summary Statistics

	Small Dataset		**Large Dataset**	
	Model 1	**Model 2**	**Model 3**	**Model 4**
Train/Validate/Test Split	1y/ 60d/ 30d	5y/ 2y/ 1y	1y/ 180d/ 180d	180d/ 180d/ 180d
+/- article split	.54	.55	.51	.51
Ave. Rho (bias)	.0050	.0065	.0011	.0013
Ave. S-Corr	.0231	.0336	.0336	.0334
Ave. S-Corr p-val	.2241	.0099	$1.4247\ e^{-15}$	$9.5452\ e^{-09}$
Ave. Accuracy	.5008	.4854	.5150	.5143
Ave. Classification Error	.4992	.5146	.4849	.4856
Ave. Sensitivity	.4762	.4256	.5228	.5267
Ave. Specificity	.5323	.5551	.5071	.5014

4.3.1. Model Assessment (Goodness of Fit)

We explored two measures to assess the goodness of fit of the model. The first was to calculate the accuracy (how often the prediction is correct), classification error (how often the prediction is incorrect), sensitivity (how often the prediction is correct if the actual value is positive) and specificity (how often the prediction is correct if the actual value is negative), all of which are traditional means of scoring a classification model. This method helped us understand how good

[4]We ran into a computational challenge when computing the co-occurrence frequencies f_j for the large dataset and were required to add min/max (.01 and .9 respectively) document frequency criteria when tokenizing the documents.

the model is at assigning a positive sentiment score to an article associated with a positive return and vice versa.

Overall, both Models 3 and 4 from the large dataset outperformed Models 1 and 2 from the small dataset, with Model 3 being the optimal model. The average accuracy score was consistently over 0.5 with the large dataset, whereas it was at 0.5 or below with the small dataset. Interestingly, the sensitivity scores for the large dataset was 0.52 for both Models 3 and 4, while the specificity scores were 0.50. On the contrary, the sensitivity scores for the small dataset were 0.47 and 0.42 for Models 1 and 2 respectively and the specificity scores were 0.53 and 0.55. These results indicate that the models trained on the large dataset were better able to pick up the signals for positive sentiment rather than negative sentiment documents, and vice versa for the small dataset models.

We looked closer at the accuracy score for the large dataset to see if it improved for the most positive and most negative sentiment articles. For the 25% most positive articles, the accuracy improved to 0.54, but went down to 0.51 for the 25% most negative articles. This further supports the claim that SESTM is better able to pick up positive sentiment signals than negative ones.

The second measure used was Spearman's rank correlation (S-Corr), which measures the correlation between the rank of the estimated sentiment and that of the true sentiment. As discussed in Section 3, a key property of the model is that the rank of the sentiment scores between $\hat{p}_{i=1}^{N}$ and $p_{i}{}_{i=1}^{N}$ is preserved, even though there is a bias associated with the tone of the article. Spearman's rank correlation is a means of measuring that relationship. The more similar the estimated rank is to the actual, the closer the Spearman's rank correlation coefficient is to 1. As n, m, N $\to \infty$, where N is the number of new articles who sentiments $p_1, ..., p_N$ are iid sampled from a continuous distribution, the estimated value of the Spearman's rank correlation coefficient should converge to one.

$$E[\text{S-Corr}(\hat{p}, p)] \to 1$$

According to the S-Corr, the rank of the predicted sentiment scores for the large dataset, Models 3 and 4, have a small, positive correlation with the actual sentiment scores. In addition, the p-value is less than 0.05 for both models, allowing us to reject the null hypothesis of no correlation. The average S-Corr coefficient is also positive for the small dataset models. However, the p-value for Model 1 is greater than 0.05, indicating that we should accept the null hypothesis and conclude that there is no correlation. For Model 2, the p-value is less than 0.05 but greater than that of the large datasets, indicating a lower level of confidence about the positive correlation. Ke et al. (2019) did not publish their average S-Corr values for their rolling window analysis, so we do not have that data point for comparison. They did perform a Monte Carlo simulation using simulated data. The positive, negative and neutral words in each article were generated using a multinomial distribution. For the returns, the sign and magnitude were simulated with a logistic regression and Student-t distributions respectively. Under these ideal, controlled conditions, the Monte Carlo simulation returned a benchmark average S-Corr of 0.85. Our large dataset showed conclusive evidence of a positive rank correlation between the estimated and actual sentiment scores, supporting the Ke et al. (2019) claim that the SESTM model preserves rank.

When compared to the small dataset, the large dataset outperformed in both accuracy and Spearman's rank correlation, pointing towards the conclusion that a large training sample is imperative to the SESTM model. Nevertheless, our training sample for Model 3 was still only $\frac{1}{10}$th the size of that used by Ke et al. (2019). We believe that the accuracy and rank correlation of the estimated sentiment scores would improve even further with a larger training set like that used by the original authors.

4.3.2. Rho (Bias on tone)

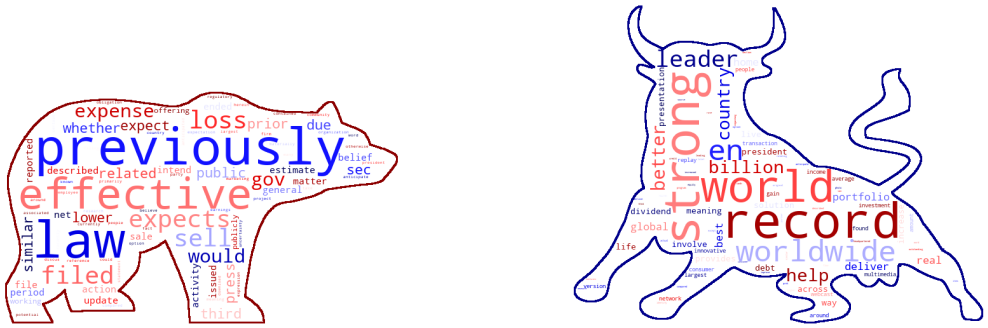
Rho is the bias on the tone of the predicted sentiment. Recall from Section 3 that tone can be determined by taking the difference of the O_+ and O_- vectors and indicates how positive or negative the sentiment is. The tone depends on the correlation between the true sentiment and the estimated sentiment, thus the bias is small when the estimation quality of p is high. The average Rho, the bias on tone, is smallest for Model 3, indicating that the estimation quality of p is highest using that model. That said, the bias has no impact on the practical usage of the estimator as we are interested in the relative sentiment, rather than the absolute sentiment.

4.3.3. Most Impactful Words

This section will focus on the optimal model, Model 3. In total, there were 84 unique positively charged words and 86 unique negatively charged words across six samples. Interestingly, the model chose 25 words, the most restrictive parameter, as the optimal $\alpha_{+/-}$ for five of the six iterations. Fig. 2 shows a word cloud of the most positive and most negative sentiment charged words. The size of the font indicates the number of times each word appears on the sentiment charged word list across all six iterations of the rolling window, with the larger font reflecting words that appear more often. Substantiating the results in Ke et al. (2019), the word list appears to be stable over time. Over six iterations, 26 words appeared in at least four iterations and five words appearing in all six. Fig. 3 below visualizes the positive and negative sentiment words that appear in four or more iterations. Given that we were only able to test on six samples, compared to the 14 samples used by Ke et al. (2019), we believe that more than six samples would be required to draw stronger conclusions about the stability of the word list.

We also calculated the frequency and the tone of the words using the O-matrix. The frequency is depicted by $\textit{Frequency} = 0.5(O_+ + O_-)$ and tone is represented by $\textit{Tone} = 0.5(O_+ - O_-)$. The words with the most positive tone, occurring in more than one sample include: increase, world, strong, billion, global, solution, presentation and income. The most negatively charged words that appeared in the top five most negative list for more than one sample include: would, loss, third, law, lower, action, expense, sale and offering. Lastly, the sentiment charged words that occurred most frequently throughout the corpus include: billion, press, filed, world, home, solution, net, sale and income.

The Loughran-Mcdonald (LM) dictionary is one of the most widely used lexicons for sentiment



(a) Negative Sentiment Words **(b)** Positive Sentiment Words

Fig. 2: Most common positive and negative sentiment words for Model 3, the optimal model. The larger font size reflects words that appeared in the sentiment charged word list more often over the six iterations.

analysis of financial text. It was created by analyzing the text of a large sample of 10-Ks during 1994 to 2008 and underscored the importance of context-specific lexicons. In their 2011 paper, Loughran and Mcdonald[10] illustrated that almost three-fourths of the words identified as negative by the widely used Harvard Dictionary, where not actually negative in financial texts. In comparing our sentiment-charged word list to the LM dictionary, only seven out of 84 positive words coincided with the LM dictionary: strong, better, best, innovative, gain, leading, improve. Of the negative sentiment words, only one out of 84 words appeared on the LM dictionary: loss. Interestingly, Ke et al. (2019) only shared one positive word and 27 negative words with the LM dictionary[5].

While the LM dictionary has a financial focus, it was trained on a corpus of 10-Ks, which are different in nature to financial news articles. 10-Ks are typically approximately 100 pages long, whereas our news articles are significantly shorter. Minimal overlap in our both our sentiment-charged word list and that of Ke et al. could indicate that the language selected to relay positive or negative information in news articles may differ slightly from that of a 10-K due to the purpose of the document. In addition, the LM dictionary was trained on documents from 1994 to 2008, whereas our large dataset spanned 2016 to 2020. The nature of business and the language used to describe it may have evolved over that time, accounting for the lack of overlap. These observations could indicate that there is uniqueness in the language used in financial news articles, compared to 10-Ks and other financial documents.

One final observation is that when we read through the words in our sentiment-charged word list, their positive and negative connotations are not as intuitively strong as those in Ke et al. (2019). In fact, only one word, 'lower' in the negative list, overlaps with Ke et al. Again, this could be due to the size of our training sample. A larger sample would drive toward stronger convergence

[5]The LM dictionary contains at total of 354 positive words and 2355 negative words.

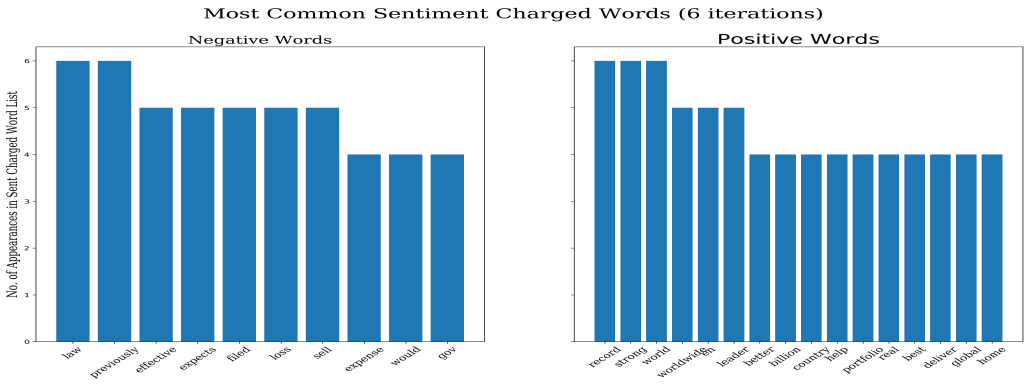


Fig. 3: Sentiment charged words appearing in four or more samples

to the most positively and most negatively charged words.

5. Other Sentiment Models

In order to benchmark the results of the SESTM model, we compared it to two commonly used approaches for textual sentiment analysis. The first is a lexicon-based approach, where we used the Sentometrics package in R to score news articles based on the Loughran & Mcdonald dictionary. For the second approach, we used Word2Vec word embeddings to embed the documents in conjunction with the XGBoost Regression algorithm to predict returns, which in this case acts as a proxy for the sentiment score.

5.1. Lexicon-Based Approach with Sentometrics

The Sentometrics package implements a dictionary-based framework to compute sentiment scores of text data. The framework used is a straightforward unigrams approach, where the computed sentiment is a weighted sum of all detected word scores as they appear in the given lexicons. The lexicon includes lists of polarized (positive and negative) words and is usually built for specific contexts. In our case, we used the Loughran & McDonald lexicon (Loughran and McDonald, 2011) [10] that was built for specific for financial text.

The advantages of the R package Sentometrics are the fast computation and aggregation of textual sentiment. Moreover, the lexicon-based approach to sentiment calculation is flexible, transparent, and computationally convenient. A drawback for the method is that it excessively relies on a collection of known and precompiled sentiment terms which may not accurately represent the corpus to be scored.

5.2. Word Embeddings with Word2Vec and XGBoost

Word2Vec is a state of the art and one of the most popular techniques to learn word embeddings using shallow neural networks [13]. The algorithm takes as an input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space. In our case we implement Word2Vec with the Gensim Library in Python, using the CBOW method, which uses the words of the context to predict the central word. Firstly, we need to preprocess the corpus and convert it into lists of words of each document [12] [3]. The Word2Vec model is trained on a collection of words. After creating the word vectors we train our data with the XGBoost Regression model in order to make predictions for the returns.

Word2Vec retains the semantic meaning of different words in a document. A great advantage of this approach is that the size of the embedding vector is very small. Each dimension in the embedding vector contains information about one aspect of the word. We do not need huge sparse vectors, unlike the bag of words and TF-IDF approaches and the resulting word vector dimension is small, saving storage and computing resources. It is fast to train compared to other techniques. This technique works for both a small amount of datasets and a large amount of datasets, so it is an easy-to-scale model. It does exceptionally well in capturing semantic similarity.

One big drawback of this approach is the inability to handle unknown or out-of-vocabulary words. If the model encounters a word it has not seen previously, it will not know how to vectorize it. Also, there are no shared representations at sub-word levels. For example, the algorithm can not understand that a new word ending in "-less" is probably an adjective indicating a lack of something. Word2vec represents every word as an independent vector, even though many words are morphologically similar, this one-to-one relationship does not solve the problem of polysemous words.

5.3. Comparison of SESTM with Other Models

Interestingly, when the S-Corr and Accuracy scores are compared across all models, the SESTM model outperformed for the small dataset, but performed similarly to the Sentometrics lexicon-based model on the large dataset. As presented in Table 3, the accuracy for SESTM was higher than that of Sentometrics for the small dataset and approximately equal for the large dataset. Of interest is that Sentometrics's specificity is higher than its sensitivity, indicating that it scored the negative sentiment documents more accurately than the positive sentiment documents. In comparison, SESTM outperformed on the positive sentiment documents. Lastly, Sentometrics, yielded a stronger rank correlation for both the large and small datasets.

The Word2Vec / XGBoost model clearly performed the worst with accuracy scores at or below

0.50 and S-Corr p-values above 0.05. While we can conclude that both Sentometrics and SESTM outperformed Word2Vec / XGBoost, we are not able to clearly conclude that SESTM out performed the lexicon based approach as Ke et al. (2019) did. Nevertheless, our results do indicate strength in the SESTM model and lend further support to our belief that a larger training sample is required to optimize the performance of the SESTM model.

Table 3: Model Comparison: Summary Statistics

	Small Dataset			**Large Dataset**		
	SESTM M1	**W2V**	**Sento.**	**SESTM M3**	**W2V**	**Sento.**
Ave. S-Corr	.0231	.0010	.0561	.0336	.0162	.0479
Ave. S-Corr p-val	.2241	.4114	0.2125	$1.4247\ e^{-15}$	.0612	$1.1147\ e^{-45}$
Ave. Accuracy	.5008	.4935	.4932	.5150	.5016	.5163
Ave. Classification Error	.4992	.5068	.4850	.4672	.4984	.4837
Ave. Sensitivity	.4762	.4925	.4490	.5228	.5016	.5092
Ave. Specificity	.5323	.4955	.5438	.5071	.5077	.5233

6. Backtesting Approach and Results

We followed a similar trading strategy to that of Ke et al (2019) to evaluate the predictive power of news article sentiment. It consists of building daily long portfolios for the most positive and negative sentiment stocks and a long-short portfolio where we buy the positive portfolio and short the negative one.

All portfolios are equal-weighted and fully invested in stocks. Equal weighting is a simple and robust method of assessing the predictive power of sentiment throughout the firm size spectrum [9]. In case we do not have enough news articles to score the number of stocks desired, we evenly split the portfolio over the stocks we have scores for.

The sentiment analysis is done for news articles published at day 0 and stocks are considered to be bought at the open price of the following day (day 1) and sold at the open price of two days ahead (day 2). We consider articles to belong to day 0 if it is posted between 9:00 am of day 0 and 9:00 am of day 1. We exclude articles from 9:00 am to 9:30 am from trading, although these news are still used for training and validation purposes. We chose to act once at the beginning of every trading session because it better represents the capabilities of an average investor.

For simulations using the large dataset, we built portfolios of 50 stocks and for portfolios using the small dataset, we built portfolios of five stocks. These adjustments are needed to account for the difference in the number of news articles available daily in each of the datasets.

Trading costs of any sort, including commissions, slippage, and the bid/ask spread, are not

considered in the analysis. Borrowing costs are also not considered for the short selling of stocks. For this reason, we take the results exclusively as a measure of the quality of the models in distinguishing outperforming stocks and do not discuss the profitability of the strategy in a realistic setting.

6.1. Backtest results

In this section, we exhibit the backtest results for the trading strategy introduced in the previous section. A model is effective when the positive sentiment portfolio outperforms its negative sentiment counterpart. We tested all three models in both small and large datasets. For the large dataset, we simulated two additional portfolios: one restricting the investable universe of stocks to the S&P 500 index composition; and another portfolio restricted to the stocks not belonging to the index.

We assess the quality of the models in predicting stock returns by analyzing the following portfolio performance measures: annualized Sharpe ratio, annualized return, annualized volatility, and maximum drawdown[6]. The Sharpe ratio measures the performance of a portfolio compared to a risk-free asset[7], after adjusting for its risk. It represents the additional return that an investor receives per unit of increase in risk. The annualized return is the geometric average of annual returns each year over the investment period. It is useful when comparing returns with different time frames. The annualized volatility is a risk metric measured as the standard deviation of a portfolio yearly return. A maximum drawdown is the maximum observed loss from a peak to a trough before a new peak is attained. It is an indicator of downside risk over a specified time period.

To ensure robustness to our analysis, we measured the performance for 12-month rolling windows (overlapping) over the backtesting period. We rolled the windows in lumps of seven days. This way, we get hundreds of different observations for the same strategy instead of comparing the results over a single period. We are especially interested in the backtest results for the long-short portfolios as they better represent the ability of a model in predicting stock returns. The better (worse) the long (short) leg of our portfolio performs, the higher is the performance of the long-short portfolio.

6.1.1. Small Dataset

The small dataset consisted of 148,167 news articles for a selection of 69 distinct stocks from January 1, 2012 to April 8, 2020. The backtesting period starts in April 3, 2013 and goes until April 8, 2020. For each portfolio, we had 319 distinct 12-month windows from which the metrics on Table 4 are averaged.

Neither of the models delivered satisfactory backtest results using the small dataset. We did not observe a distinct behavior between the positive and negative sentiment stocks portfolios and

[6] All metrics were calculated using the Empyrical package for Python - https://github.com/quantopian/empyrical

[7] We considered a constant annual risk-free rate of 2.00% for the calculations presented in this paper

all of the long-short portfolios had negative average returns over the rolling windows.

Table 4: Average performance metrics for the small dataset

	Sharpe Ratio	Ann. Ret	Ann. Vol	Max DD
Positive sentiment				
SESTM	0.62	13.0%	21.2%	-17.0%
Sentometrics	0.76	14.2%	19.3%	-14.5%
Word-Embeddings	0.81	17.5%	20.6%	-16.3%
Negative sentiment				
SESTM	0.80	17.3%	21.3%	-16.0%
Sentometrics	0.94	19.8%	19.8%	-15.1%
Word-Embeddings	0.93	20.2%	21.0%	-17.5%
Long-Short				
SESTM	-0.29	-5.3%	19.3%	-20.1%
Sentometrics	-0.39	-4.7%	14.7%	-17.4%
Word-Embeddings	-0.37	-1.3%	18.5%	-23.2%

We argue that none of the models performed well with the small dataset partially because the training data was small[8], but also because we did not have sufficient stocks to pick when executing the trading strategies. Even though we had articles for over 60 distinct stocks in total, in many days there were less than 10 stocks with sentiments scores. This limitation critically affects the capacity of the models to pick stocks with strong opposing sentiments. Also, most of the stocks were large caps concentrated in the technology and financial sectors, limiting, even more, the capacity of the strategy to select stocks with different behaviors.

6.1.2. Large Dataset

The large dataset consisted of 2,883,935 news articles for a selection of 7,324 unique stocks from January 1, 2016 to April 30, 2020. The backtesting period starts in July 6, 2017 and goes until April 30, 2020. For each portfolio, we had 96 distinct 12-month windows from which the metrics on Table 5 are averaged.

The backtest portfolios using the models trained with the large dataset performed better than those with the small dataset. The results for both the SESTM model and Sentometrics look good, with a clear difference in performance between the positive and the negative sentiment portfolios. The SESTM model performing better than a dictionary-based method is in line with the results

[8]This argument is supported by the improvement in the SESTM model metrics displayed on Table 3 when trained with the small dataset to when it was trained with the large dataset.

found by Ke et al (2019). When comparing the Sharpe ratio of our backtest to those reported by Ke et al (2019)[9], our implementation of the SESTM performed slightly worse, however, we recall that the measures are not comparable across papers because of different test time ranges, universes of stocks and amount of training data. It is important to point out that they trained their model in ten years of data, while we did it in a one-year data.

Table 5: Average performance metrics for the large dataset

	Sharpe Ratio	**Ann. Ret**	**Ann. Vol**	**Max DD**
Positive sentiment				
SESTM	1.72	39.2%	20.3%	-18.2%
Sentometrics	0.85	18.2%	19.5%	-22.2%
Word-Embeddings	-0.35	-9.1%	24.3%	-30.5%
Negative sentiment				
SESTM	-1.18	-22.4%	21.5%	-35.6%
Sentometrics	-1.05	-19.9%	21.1%	-33.1%
Word-Embeddings	-0.50	-9.6%	18.9%	-27.9%
Long-Short				
SESTM	3.62	79.8%	15.8%	-6.7%
Sentometrics	2.51	50.1%	14.6%	-12.8%
Word-Embeddings	-0.16	0.2%	10.2%	-10.2%

The word-embedding long-short portfolio performed poorly and indicates the word-embedding model was not able to successfully predict the outperforming stocks. This is evidenced by the similar average returns in the positive and negative sentiment legs.

We also plot the cumulative returns graphs over the backtesting period for the portfolios to visually represent the results. We note a clear upward trend of the long-short portfolio, indicating the model was able to predict a group of stocks that would outperform.

[9]The SESTM model achieved a Sharpe ratio of 4.29 for the equal-weighted long-short portfolio.

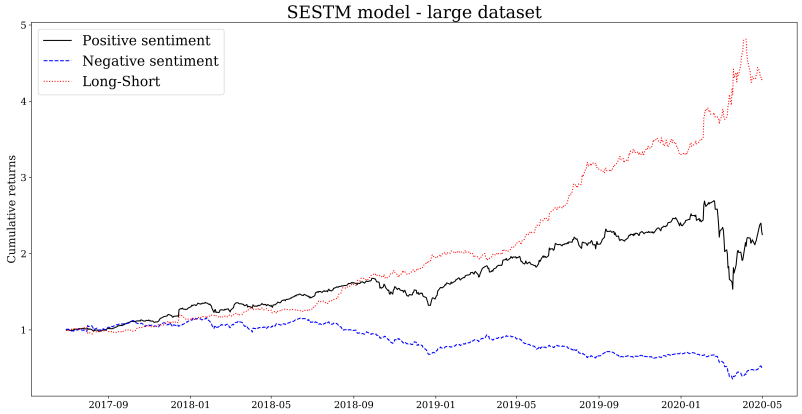


Fig. 4: Cumulative returns for the SESTM model

A relevant observation across both the SESTM and the Sentometrics simulated portfolios is that not only the long-short strategy avoided big drawdowns during the 2020 stock market crash, but it delivered big gains. Even though the positive legs had a big drawdown of close to 40%, the negative sentiment legs had larger drawdowns of close to 70%. A potential economical explanation for this finding is the positive association of the cross-section equity returns dispersion and volatility. In a moment of high volatility of the stock market the difference between the top and the worst performance stocks tends to be larger.

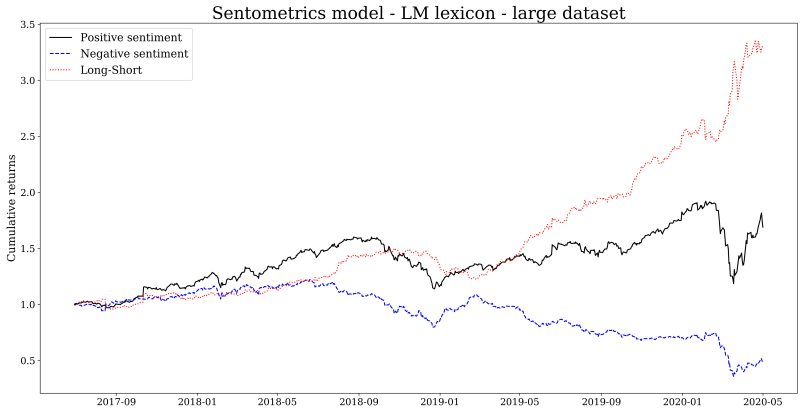


Fig. 5: Cumulative returns for the Sentometrics model

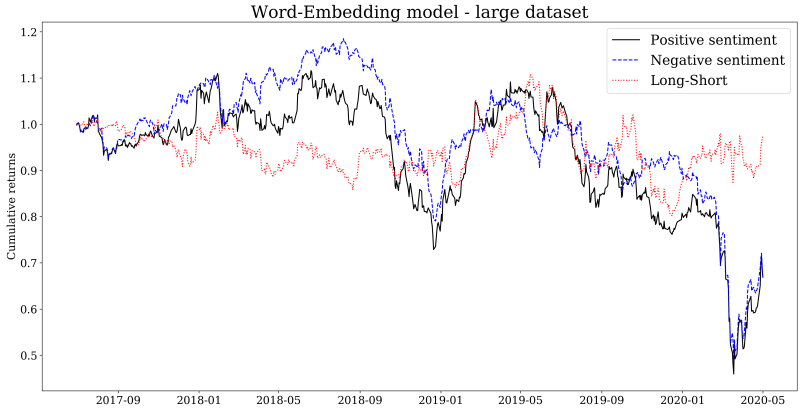


Fig. 6: Cumulative returns for the Word-embedding model

6.1.3. Large Cap. vs Small-to-Mid Cap Stocks

As an experiment of the effectiveness of the backtest for stocks across different market capitalization and liquidity levels, we ran two additional portfolios: one restricted to stocks in the S&P 500 index composition and another excluding those stocks. Although it is not a perfect metric, we use the S&P 500 companies as a proxy for larger market capitalization and higher liquidity stocks in our investable securities universe.

Table 6: Average performance metrics for portfolios exclusively with S&P 500 stocks

	Sharpe Ratio	**Ann. Ret**	**Ann. Vol**	**Max DD**
Positive sentiment				
SESTM	0.76	12.9%	16.4%	-15.0%
Sentometrics	0.85	14.4%	16.1%	-15.8%
Word-Embeddings	0.76	12.7%	16.4%	-15.5%
Negative sentiment				
SESTM	0.76	13.0%	17.8%	-19.3%
Sentometrics	0.15	1.9%	18.2%	-19.6%
Word-Embeddings	0.66	11.8%	18.0%	-19.2%
Long-Short				
SESTM	-0.25	0.0%	9.8%	-12.3%
Sentometrics	0.90	11.8%	10.1%	-7.0%
Word-Embeddings	-0.16	0.2%	10.2%	-10.2%

The results are in line with the findings of Ke et al (2019) that news articles sentiment is a stronger predictor of future returns to small cap. stocks, all else equal. The Sharpe Ratios for the non-S&P long-short portfolios were much higher for the SESTM and Sentometrics models at 3.88 and 3.0 respectively, than the S&P portfolios with ratios of -0.25 and 0.90 respectively. The potential economical explanations for this observation involve small stocks receiving less investor attention and therefore responding more slowly to news; the underlying fundamentals of small stocks are more uncertain and thus it requires more effort to process news into actionable price assessments; or small stocks are less liquid and thereby require a longer time for trading to happen and incorporate information into prices.

Interestingly, the SESTM model outperformed Sentometrics for the non-S&P 500 portfolio, but Sentometrics yielded better results for the S&P 500 portfolio. Additional exploration into semantics or signals in articles of S&P vs. non-S&P 500 stocks would be required in order to explain the difference.

Table 7: Average performance metrics for portfolios built exclusively with non-S&P 500 stocks

	Sharpe Ratio	**Ann. Ret**	**Ann. Vol**	**Max DD**
Positive sentiment				
SESTM	1.62	39.5%	21.8%	-19.3%
Sentometrics	1.10	26.8%	21.5%	-21.9%
Word-Embeddings	-0.27	-8.0%	26.0%	-31.2%
Negative sentiment				
SESTM	-1.70	-31.8%	22.7%	-41.7%
Sentometrics	-1.38	-26.6%	22.5%	-38.2%
Word-Embeddings	-0.75	-14.3%	20.0%	-30.9%
Long-Short				
SESTM	3.88	104.4%	18.3%	-8.1%
Sentometrics	3.00	72.9%	17.7%	-13.9%
Word-Embeddings	0.30	7.2%	22.3%	-16.2%

In this scenario, we do not see the same effectiveness observed in the simulation with all stocks for the SESTM model. Sentometrics seems to be the only model slightly effective in this settings, however still not as good as the simulations considering all stocks.

Figures 7, 8 and 9 below provide a visual representation of the difference in backtesting performance across our proxy large cap and small-to-medium cap stock portfolios.

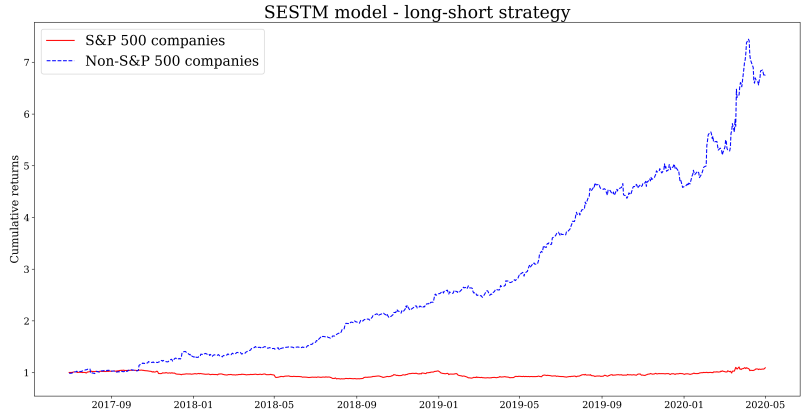


Fig. 7: Cumulative returns for SESTM - segregated by S&P 500/Non-S&P 500 companies

From the annualized volatilities of the simulated portfolios in Tables 6 and 7, we also infer that Non-S&P 500 companies are more volatile than those stocks belonging to the index. This inference might further explain the over-performance of the trading strategy applied to non-S&P 500 stocks, as noted by Ke et al (2019) that, "while news about low volatility firms is fully impounded in prices after one day of trading, it takes three days for news to be fully reflected in the price of a high volatility stock."

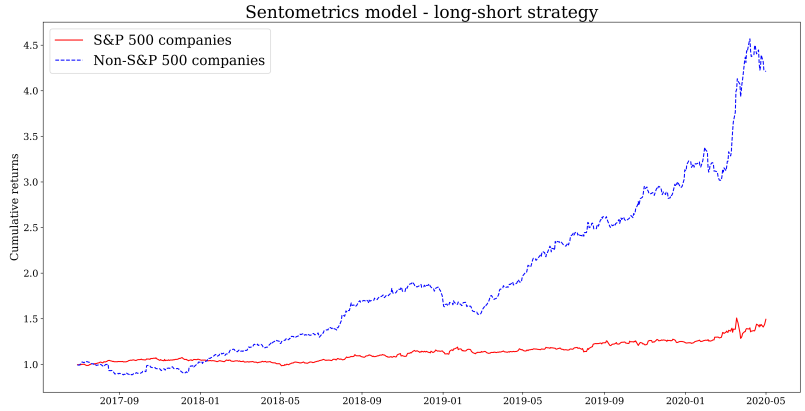


Fig. 8: Cumulative returns for Sentometrics - segregated by S&P 500/Non-S&P 500 companies

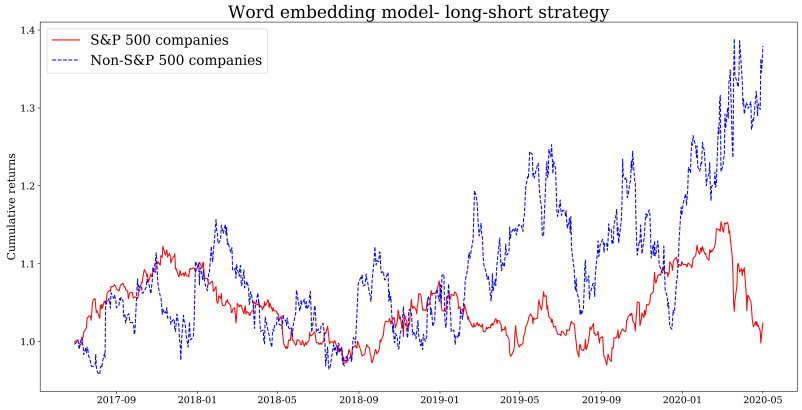


Fig. 9: Cumulative returns for Word embedding- segregated by S&P 500/Non-S&P 500 companies

7. Conclusion

Overall, we found that the SESTM model proposed by Ke et al. (2019) has merit as a supervised approach to predict sentiment scores conditioned to stock returns. We were able to replicate most of the findings by the original authors, even though the dataset available to us was only a fraction of the size of theirs.

We agree that SESTM model is "white-box" and interpretable. Each step of the algorithm is transparent and explainable. Once the model is trained, we were able to identify the sentiment charged words and decompose them into their tone and frequency weights. In terms of computational cost, the process of scoring the documents is computationally efficient, however calculating the co-occurrence frequencies f_j for the large dataset was computationally challenging. We were forced to put min and max document frequency restrictions on the word list before that step in order to reduce it to a computationally efficient size.

Our experience with implementing the methodology and algorithms described in the original paper showed the importance of having an extensive amount of data to properly train the model, at least for low signal to noise ratio data such as financial returns. After training four versions of the model with two datasets of different sizes, we found that the models trained on the large dataset largely outperformed those trained on the small dataset.

By backtesting a systematic trading strategy using only the article sentiment scores to buy and sell stocks, we were able to assert the SESTM model's competence in a practical application. The backtest results were in line with the financial literature, specifically around how financial news is incorporated into stock prices. The simulated long-short portfolios delivered significant average annual returns over the rolling window analysis, with the caveat that transaction costs were not

considered. We also substantiated Ke et al.'s observation that news is incorporated more slowly into the price of small cap stocks than that of large cap stocks, observing that both the SESTM and Sentrometrics models significantly outperformed on the non-S&P 500 portfolios compared to the S&P 500 portfolios.

When compared to the Loughram-McDonald (LM) dictionary-based model (Sentometrics), SESTM slightly outperformed the benchmark Sentrometrics model. These results, when considered with the summary statistics of the two models, led us to believe that the SESTM model is a viable alternative to a traditional dictionary-based method when data availability is high. Our implementation of the word-embedding model, however, significantly under-performed compared to the other two models.

Regarding the sentiment-charged words identified by the SESTM model, we find that they were not as naturally intuitive as those in the LM dictionary or the Ke et al. word list. Given that Ke et al.'s was more intuitive, we believe that training on a larger dataset could drive towards a more intuitive word list. One final comment is that we also saw a difference in the similarity of our word list to the LM dictionary compared to Ke et al. Ke at al's negative word list intersected more with the LM dictionary, whereas our positive word list intersected more. This, combined with our observation that our implementation of the SESTM model scored the positive articles more accurately than the negative ones, could be due to the distinctness of our dataset, as it only overlapped Ke et al.'s by 18 months.

For future work, robustness could be added to the analysis by (i) incorporating more data to allow larger training sets and more testing periods; (ii) combining the sentiment-charged word list with the most common words from the LM dictionary to incorporate known signals; (iii) considering transaction and implementation costs to the trading strategy backtest; (iv) experiment with implementing the model for a different prediction task (e.g., predicting volatility).

References

[1] Werner Antweiler and Murray Z Frank. "Is all that talk just noise? The information content of internet stock message boards". In: *The Journal of finance* 59.3 (2004), pp. 1259–1294.

[2] David Ardia et al. "The R package sentometrics to compute, aggregate and predict with textual sentiment". In: *Aggregate and Predict with Textual Sentiment (November 9, 2017)* (2017).

[3] V Kishore Ayyadevara. *Pro Machine Learning Algorithms*. Springer, 2018, pp. 167–178.

[4] Johan Bollen, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market". In: *Journal of computational science* 2.1 (2011), pp. 1–8.

[5] Alfred Cowles 3rd. "Can stock market forecasters forecast?" In: *Econometrica: Journal of the Econometric Society* (1933), pp. 309–324.

[6] Jianquin Fan and Jinchi Lv. "Sure independence screening for ultrahigh dimensional feature space". In: *Journal of the Royal Statistical Society* 70.5 (2008), pp. 849–911.

[7] Matthew Gentzkow, Bryan Kelly, and Matt Taddy. "Text as data". In: *Journal of Economic Literature* 57.3 (2019), pp. 535–74.

[8] Sheng Tracy Ke and Minzhe Wang. "A new svd approach to optimal topic estimation". In: *Technical report* (2017).

[9] Zheng Tracy Ke, Bryan T Kelly, and Dacheng Xiu. *Predicting returns with text data*. Tech. rep. National Bureau of Economic Research, 2019.

[10] Tim Loughran and Bill McDonald. "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks". In: *The Journal of Finance* 66.1 (2011), pp. 35–65.

[11] Asaf Manela and Alan Moreira. "News implied volatility and disaster concerns". In: *Journal of Financial Economics* 123.1 (2017), pp. 137–162.

[12] Ahmed Menshawy. *Deep Learning By Example: A hands-on guide to implementing advanced machine learning algorithms and neural networks*. Packt Publishing Ltd, 2018, pp. 285–309.

[13] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[14] Paul C Tetlock. "Giving content to investor sentiment: The role of media in the stock market". In: *The Journal of finance* 62.3 (2007), pp. 1139–1168.