Stock portfolio selection using learning-to-rank algorithms with news sentiment

Qiang Song[a], Anqi Liu[a], Steve Y. Yang[a,*]

[a]*Financial Engineering Division, Stevens Institute of Technology, 1 Castle Point on Hudson, Hoboken, New Jersey, USA 07030*

Abstract

In this study, we apply learning-to-rank algorithms to design trading strategies using relative performance of a group of stocks based on investors' sentiment toward these stocks. We show that learning-to-rank algorithms are effective in producing reliable rankings of the best and the worst performing stocks based on investors' sentiment. More specifically, we use the sentiment shock and trend indicators introduced in the previous studies, and we design stock selection rules of holding long positions of the top 25% stocks and short positions of the bottom 25% stocks according to rankings produced by learning-to-rank algorithms. We then apply two learning-to-rank algorithms, ListNet and RankNet, in stock selection processes and test long-only and long-short portfolio selection strategies using 10 years of market and news sentiment data. Through backtesting of these strategies from 2006 to 2014, we demonstrate that our portfolio strategies produce risk-adjusted returns superior to the S&P500 index return, the hedge fund industry average performance - HFRIEMN, and some sentiment-based approaches without learning-to-rank algorithm during the same period.

Keywords: Learning-to-rank, Stock portfolio selection, Long-short strategy, Financial news sentiment, Trading strategy

*Corresponding author

Email addresses: qsong@stevens.edu (Qiang Song), aliu@stevens.edu (Anqi Liu), steve.yang@stevens.edu (Steve Y. Yang)

1. Introduction

Impacts of investors' sentiment to financial market have been well documented in a number of behavioral finance studies [3, 33, 12, 25, 9]. However, since investors' sentiment is an abstract concept, researchers have put a lot of effort in finding adequate and reliable proxies to represent its underlying mechanism. These proxies include mutual fund flows, closed-end funds prices, news and social media messages aimed to extract investors' attitude towards financial market [20, 33]. In recent years, due to its broad digital accessibility, news media has been increasingly becoming a critical source of information that supports investors' investment decisions. The arrival of news continually updates investors' understanding and knowledge of the market and influences investor sentiment as a result. Most notably, the primary financial data vendors, such as Bloomberg and Thomson Reuters, now publish news analytic solutions to traders and investors in an almost real-time fashion. According to the enhanced news feeds delivery, many recent studies have focused on parsing news articles to calibrate investor sentiment through text mining and machine learning techniques [29, 24, 37].

In this paper, we employ a machine learning ranking method, learning-to-rank algorithms, to construct equity portfolios based on news sentiment. Many machine learning algorithms have been used for financial market prediction and trading strategy development [11, 17, 19, 23, 34], these efforts have been mostly focused on constructing portfolios based on return forecasting. We propose a stock portfolio construction approach from the viewpoint of ranking investors' relative views on stocks' performance. Learning-to-rank is a class of algorithms that apply supervised machine learning approaches to solve ranking problems, and it is a task to automatically construct a ranking model using training data, such that the model can sort new objects according to their degrees of relevance, preference, or importance [22]. We argue that many investment or trading decisions can be naturally cast into a ranking problem. For example, to identify which assets will outperform in the future. Making an investment decision is essentially to rank all the assets in the investable universe and buy the top ones or short the bottom ones. With increasing data availability, it is natural to leverage machine learning algorithms for such ranking problems. Machine learning algorithms are proven to be effective in fitting parameters automatically, avoiding over-fitting, and being capable of combining multiple inputs. Ranking investors' sentiment hence provides a natural way to select stocks based on the "portrayed performance" in news media.

The advantages of using learning-to-rank algorithms in portfolio selection are twofold. First, unlike the traditional machine learning algorithms that predict a value for one input based on past information, learning-to-rank provides ranks for a group of inputs. In other words, learning-to-rank targets on relative orders instead of absolute values. This property is particularly useful for investment applications. Secondly, we can combine different performance indicators, and the parameters tuning can be done automatically. Traditionally, a ranking model is built without training. Items are ranked by either relevance or importance, such as Boolean model [2], BM25 [28] and PageRank [27]. These conventional ranking models require parameter tuning and face over-fitting problems. In addition, combining different models to get better results is not straightforward. Learning-to-rank algorithms are particularly suitable for investment decisions. Unlike traditional machine learning algorithms, which only make one prediction at a time, learning-to-rank predicts a ranked list. This approach presents a relative performance among stocks, and it is more reliable than an absolute performance forecast. For the majority of the money managers, their task is not to achieve absolute return, but to outperform a specific benchmark. Moreover, there are often a plethora of indicators for predicting stock returns, and sometimes they give conflicting signals. The learning-to-rank algorithms can automatically combine a range of indicators and weigh them to produce optimal results.

The primary contribution of this study is to demonstrate the outperformance and robustness of a relative performance based portfolio construction method: learning-to-rank algorithms in designing trading strategies using a relative view of investors' sentiment. It bridges the gap of predicting relative performance for a group of stocks through multiple information sources. We argue that learning-to-rank algorithms are effective in producing reliable ranking models to predict the best and the worst performing stocks based on investor sentiment and market information. In a previous study, we designed sentiment shock and trend indicators to investigate firm-specific stock price movements and generate trading signals for individual stocks. This trading strategy is based on the hypothesis that ranking of news sentiment reflects expected returns during the near future. In this study, we use the sentiment shock and trend indicators introduced in previous studies [31, 38] to develop stock selection rules of holding long positions of the top 25% stocks and short positions of the bottom 25% stocks according to the stock rankings produced by learning-to-rank algorithms. In the experiments of portfolio strategies, we apply ListNet and RankNet in stock selection processes and test long-only and long-short strategies. Applying backtesting of the models for the period from 2006 to 2014, we show that the selected portfolios using our learning-to-rank methods have superior profitability to the S&P500 index. Moreover, the long-short strategies produce robust Sharpe ratio in both high volatility and low volatility regimes. This study demonstrates two key features of the proposed methods. First, learning-to-rank algorithms produce accurate predictions of expected return rankings, and the proposed stock selection approach works robustly under different financial market conditions. Secondly, the sentiment indicators [31, 38] support ranking predictions consistently in reflecting individual stock's future performances under different market conditions.

The rest of the paper is organized as follows. In the next section, we review existing studies on sentiment analysis and application of machine learning methods to investment decisions. Section 3 introduces sentiment shock and trend indicators, ListNet and RankNet algorithms, and the stock selection approach. Section 4 presents data sources of market information and news sentiment along with their statistical relationships. Section 5 describes an application of portfolio strategy with the proposed stock selection process. Section 6 discusses the key findings of the portfolio management experiments based on learning-to-rank technique. Section 7 concludes the discussion and proposes some future work.

2. Background and related literature

2.1. Trading with financial news sentiment

Financial sentiment has been widely explored in both academic and industrial research. To evaluate investor sentiment, researchers have used a variety of information sources to form sentiment proxies. Financial news has been one primary source of investor sentiment analysis. In general, research on financial news impacts to markets targeted on explaining two questions: 1) Does news information lead financial market activities? 2) Can special patterns of news sentiment form indicators that provide reliable prediction about subsequent market price movements? A series of studies suggested empirical evidence of statistical relationships between news and financial markets. By exploring different market features, news impacts can be described from three perspectives. First, news information is associated with subsequent market return. By analyzing millions of messages from Yahoo! Finance and Raging Bull, Antweiler and Frank [1] documented that the number of posts has significant correlation with market return. Second, financial news indicates market volatility level. In a previous study, we used linear regression to demonstrate that abnormal news sentiment has significant predictive power of implied volatility of S&P 500 index [38]. Third, news sentiment impacts trading volume. Antweiler and Frank [1] discussed this question from the view point of disagreement in news and confirmed that fluctuations of sentiment polarity raise trading volume.

Findings of news impacts on financial markets led to further studies of sentiment-based algorithmic strategies. Tetlock [33] developed a trading strategy applying firm-specific news content of a previous trading day and concluded that the negative content in media information provides significant predictive power in risk-adjusted returns. In a similar study, Khadjeh Nassirtoussi et al. [18] implemented a multi-layer dimension reduction algorithm on news headlines to predict intraday direction of the USD-EUR pair and achieved an accuracy of 83%. Mitra et al. [26] incorporated both market information and news sentiment to estimate equity portfolio volatility. In another study, Healy and Lo [14] designed a real-time news analytics framework and used Thomson Reuters News Scope data to manage investment risks and returns. Leinweber and Sisk [21] justified the predictability of news sentiment to market returns and designed sentiment-based portfolio strategies.

2.2. Machine learning algorithms for trading strategies

Machine learning algorithms are widely used for financial market prediction and trading strategies, especially for automated trading strategies. These applications can be categorized into three types. The first type of application predicts future asset prices or returns. Generally, regression [23] and neural network [17] algorithms are used in this type of strategies. The shortcoming of this approach is high error rates due to the difficulties in pinpointing future assets value according to chaotic financial market data. The second type is to predict price movement directions utilizing classification algorithms such as SVM [19] and decision trees [34]. These approaches usually have high prediction accuracy. But good predictions do not necessarily lead to high profitability. For example, a model may predict correctly on small gains but incorrectly on large losses, which results in large down-side risk. The third type is rule-based optimization. It aims at determining optimal combinations of trading indicators (e.g., trading indicators such as technical indicators, fundamental indicators and macroeconomic indicators) and parameters. Optimization algorithms that have been explored include genetic programming [10] and reinforcement learning [11].

2.3. Learning-to-rank algorithms and its applications

Learning-to-rank is a class of algorithms that apply supervised machine learning approaches to solve ranking problems. Originally, this technique is designed for information retrieval problems such as document retrieval and collaborative filtering [22]. In recent years, the development of learning-to-rank technique is driven by applications for online searching engine and recommendation system. Learning-to-rank algorithms can be categorized into three groups based on their inputs representations and loss functions [22]. The first one is pointwise approach. This approach predict exact ranking score for each input. It can be modeled with regression, classification or ordinal regression. Pointwise approach includes Prank [8], OC SVM [30]. The second group is pairwise approach. In this approach, ranking is transformed into binary classification. To maximize the ranking consistency, it is trained based on the relative order of each pair of input. The advantage of pairwise approach is to directly apply existing classification algorithms in training. However, the group structure is ignored in this approach. Pairwise approach includes RaningSVM [15], RankBoost [13], RankNet [4], LambdaMART [35], and Lambda Rank [5]. The last group is listwise approach. This is a more direct approach for the ranking problem. It takes the ranking lists as instances for both learning and prediction. Listwise approach includes ListNet [6], AdaRank [36], SoftRank [32] and SVM MAP [39]. According to previous studies, listwise approach has the best performance, followed by pairwise approach [22].

3. Methodology

3.1. News sentiment indicators

The correlation between news sentiment and market return indicates that sentiment can be an indicator to market movements. However, this time series property may not be effective on cross-sectional analysis of a group of assets. Absolute sentiment scores do not accurately represent relative performance of different stocks. Therefore, we examine the structure of sentiment time series and design two sentiment indicators. The sentiment shock scores normalize firm-specific sentiment based on the deviation from mean and make it comparable across different stocks. The sentiment trend scores remove drift and make the time series stationary.

Sentiment shock score: Sentiment shocks are spikes in time series which are usually caused by the releasing of unexpected macroeconomic data, financial reports, and corporate actions. To capture the shocks, we define the score as the level that current sentiment deviates from previous average value (see Equation 1).

$$S_{\text{shock}}(t) = \frac{S_{\text{sentiment}}(t) - \mu(t-N, t-1)}{\sigma(t-N, t-1)} \tag{1}$$

where $S_{\text{sentiment}}(t)$ is the sentiment score on week t, N is the look-back time window, and $\mu(t-N, t-1)$ and $\sigma(t-N, t-1)$ are average sentiment and standard deviation of sentiment during $t-N$ to $t-1$.

Sentiment trend score: Sentiment trend is the aggregated sentiment change during a time period. A series of good or bad news may cause long-term upswing or downswing in sentiment and lead to a strong impact on asset price movements. In such cases, sentiment changes can be more informative than absolute sentiment scores. We define sentiment trend score as the sum of deltas of sentiment through a period of time (see Equation 2).

$$\begin{aligned} S_{\text{trend}}(t) &= \textstyle\sum_{i=t-N}^{t-1} \Delta S_{\text{sentiment}}(i) \\ \Delta S_{\text{sentiment}}(i) &= S_{\text{sentiment}}(i) - S_{\text{sentiment}}(i-1) \end{aligned} \tag{2}$$

where $S_{\text{sentiment}}(t)$ is the sentiment score on week t, N is the look-back time window.

3.2. Learning-to-rank algorithms

Learning-to-rank algorithms aim at estimating ranks of a list of items. They apply traditional machine learning techniques as base scoring models. Then the items are ranked by scores. We apply two learning-to-rank algorithms, a pairwise approach RankNet [4] and a listwise approach ListNet [6]. Three key components of these algorithms are neural network model, cross-entropy loss function, and gradient descent optimization process (see Figure 1).

Query and data labeling: Learning-to-rank algorithms are supervised learning approaches. The training data includes a series of queries. Each query

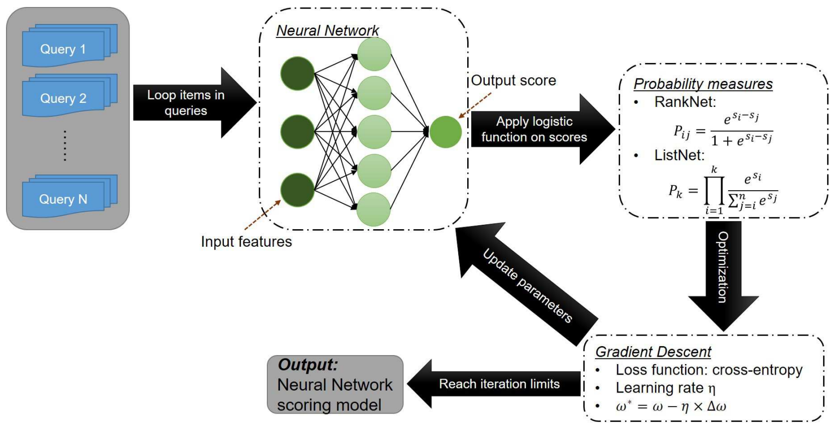


Figure 1: ListNet and RankNet algorithms flowchart. *Notes*: This figure shows optimization process of RankNet and ListNet. These two algorithms use neural network as the base model, cross entropy as the loss function, and gradient descent as the optimization rule.

contains a list of items with feature vectors and rank labels. The number of items varies in different queries and ranks are independent among queries.

To transfer the stock selection to a ranking problem, we form one query on each week and label it by rankings of weekly returns. The goal is to predict return rankings for the stocks. We analyze the stock universe and record six features for each stock: 1) sentiment shock score, 2) sentiment trend score, 3) 1-week leading return, 4) 1-month leading return, 5) 1-week leading average sentiment, and 6) 1-month leading average sentiment. These features contain information about investor sentiment and previous market performance. On each week, stocks are labeled by ranks of their following 1-week returns. We set four labels based on return quartiles. The top 25% of stocks get the highest score four and the bottom 25% of stocks are assigned the lowest score one.

Neural network model: RankNet and ListNet use neural network model to estimate scores of items in each query. In the rank prediction, a higher score leads to a higher rank (see Equation 3).

$$s_{n_i} > s_{n_j} \Rightarrow n_i \rhd n_j \tag{3}$$

where n_i is the ith item in query n, and s_{n_i} is the score of item n_i.

In the neural network model, loss function is defined as cross-entropy between the probability distribution of predicted ranks and target ranks. RankNet and ListNet use different approaches to transfer scores into probability measures. RankNet uses a pair of items as a training instance. To get probability for each training instance, it adopts logistic function on the difference of two scores (see Equation 4). On the other hand, ListNet uses a rank list of a series of items as a training instance and measures top k probability (see Equation 5).

$$P_{ij} = P(i \triangleright j) = \frac{\exp(s_i - s_j)}{1 + \exp(s_i - s_j)} \tag{4}$$

where s_i is the score of item i.

$$P(\mathcal{R}_k(i_1, i_2, ...i_k)) = P(i_1 \triangleright i_2 \triangleright ... \triangleright i_k) = \prod_{j=1}^{k} \frac{\exp(s_{i_j})}{\sum_{l=j}^{n} \exp(s_{i_l})} \tag{5}$$

where s_{i_j} is the score of item i_j.

Cross-entropy is applied as the loss function for RankNet and ListNet (see Equation 6). According to these probability measurements, we can determine and minimize the value of the loss function to obtain parameters for neural network model. In this process, predicted rank scores are calculated by the neural network model so that the value of the loss function is determined by the parameters of the neural network.

$$\begin{aligned} C_{RankNet} &= -\sum \bar{P}_{ij} \times \log P_{ij} + (1 - \bar{P}_{ij}) \times \log(1 - P_{ij}) \\ C_{ListNet} &= -\sum \bar{P}(\mathcal{R}_k(i_1, ...i_k)) \log P(\mathcal{R}_k(i_1, ...i_k)) \end{aligned} \tag{6}$$

where $\bar{P}_{ij}$ and $\bar{P}(\mathcal{R}_k(i_1, ...i_k))$ are target probabilities, P_{ij} and $P(\mathcal{R}_k(i_1, ...i_k))$ are predicted probabilities.

Gradient descent optimization: In RankNet and ListNet, the neural network parameters are determined by gradient descent optimization. On every iteration, the gradient of loss is calibrated and parameters are updated (see Equation 7). The learning rate η is the key controller of this process. A higher learning rate determines faster convergence. However, an overly aggressive learning rate may lead to over-fitting very quickly.

$$\omega_i^* = \omega_i - \eta \times \Delta\omega_i \tag{7}$$

where ω_i^* and ω_i are the updated parameter and the current parameter, $\Delta\omega_i$ is the gradient of parameter ω_i, and η is the learning rate.

3.3. Learning-to-rank for trading

In investment strategies, we target on predicting the return rankings. The key point is to transform predicted rankings to trading signals. In this study, we follow the rule of "long the top and short the bottom". As we label stocks into four tiers according to weekly returns, we process the same labeling framework based on neural network scores. The top 25% scores are labeled four and the bottom 25% scores are labeled one. Then we can long the 25% stocks in the top tier and short the 25% stocks in the bottom tier.

4. Data

Market data and financial news sentiment data are obtained from Bloomberg terminal and Thomson Reuters News Analytics (TRNA) respectively. We collect data from January 2003 to December 2014.

4.1. Financial news sentiment

Thomson Reuters News Analytics is a structured database with over 80 metadata fields about financial news. It provides sentiment for each company mentioned in each news article. The sentiment is quantified as positive, negative and neutral probabilities so that we can customize the formula for our sentiment score. The fields we used for sentiment calibration in this study are listed below.

- datetime: The date and time of the news article.
- price: Reuters Instrument Code (RIC) of the stock for which the sentiment scores apply.
- sentiment: The predominant sentiment value for a piece of news with respect to a stock (i.e., 1 for positive sentiment, 0 for neutral and -1 for negative sentiment).

- pos, obj, neg: Positive, neutral, and negative sentiment probability (i.e., $\text{pos} + \text{obj} + \text{neg} = 1$).

- relevance: A real-valued number between 0 and 1 indicating the relevance of a piece of news to a stock. One news article may refer to multiple stocks. The stock with more mentions will be assigned a higher relevance.

To evaluate the sentiment score for each stock mentioned in each news article, we calculate the expected value of sentiment probabilities adjusted by relevance value (see Equation 8). The weekly financial news sentiment is the average of all sentiment scores on news published within the week.

$$S_{\text{sentiment}} = \text{relevance} \times (1 \times \text{pos} + 0 \times \text{obj} + (-1) \times \text{neg}) \tag{8}$$

4.2. Stock universe

We define a stock universe for portfolio management by the following two steps:

1). Select stocks with high liquidity. We include top 1000 stocks with highest average trading volume.

2). Filter out stocks with few news mentions. We exclude stocks with less than one news article per week.

We extract a list of 512 stocks. Figure 2 shows the number of stocks in the ten Global Industry Classification Standard (GICS) sectors which comprise our stock population (512). Comparing with S&P 500 index, our selected stock universe is a good approximation of the large-cap market.

4.3. Summary statistics of sentiment and market data

According to the filtered stock universe, we group associated news articles by week. Then we summarize statistics about the number of news articles per week and weekly average sentiment (see Table 1).

In addition, we aggregate monthly news sentiment data and compare with the S&P500 index monthly return. The average news sentiment is positively

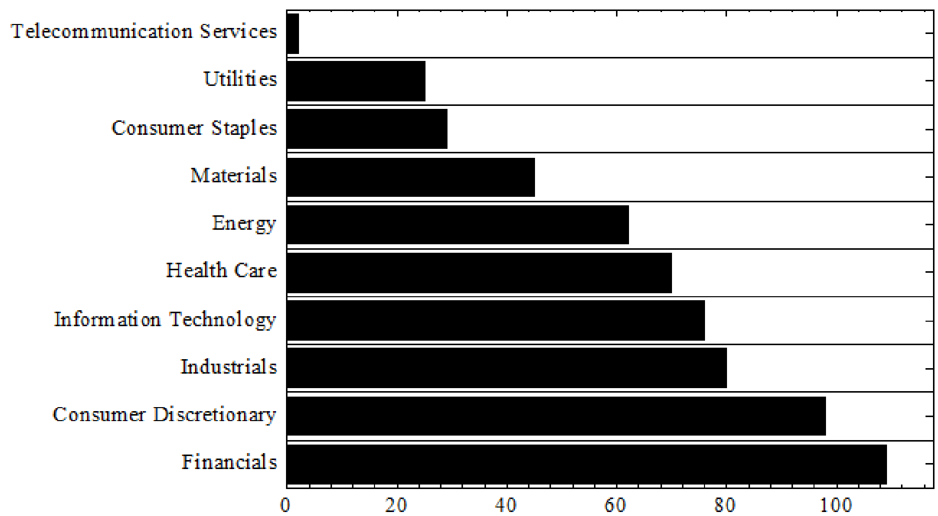


Figure 2: Number of stocks by GICS sectors. *Notes*: The stock universe contains 512 stocks and covers 10 GICS sectors. This figure shows the number of stocks in each sector.

Table 1: Statistics of news sentiment data

	Mean	**Median**	**Std.**	**Max.**	**Min.**	**5%**	**95%**
Number of news articles	5.18	2	11.69	830	0	0	22
Avg. sentiment	0.09	0.00	0.24	0.83	-0.78	-0.20	0.60

Notes: This table summarizes mean, median, standard deviation, maximum, minimum, 5% quantile and 95% quantile values of the number of news articles per week and average weekly news sentiment.

correlated with market return, while the number of news articles is negatively correlated with market return (see Figure 3). The correlation coefficients are 0.21 and -0.14 respectively. These results are consistent with empirical findings of Antweiler and Frank [1] in which the authors mentioned that most of growth in news-volume is caused by bullish messages. Through lead-lag analysis, we

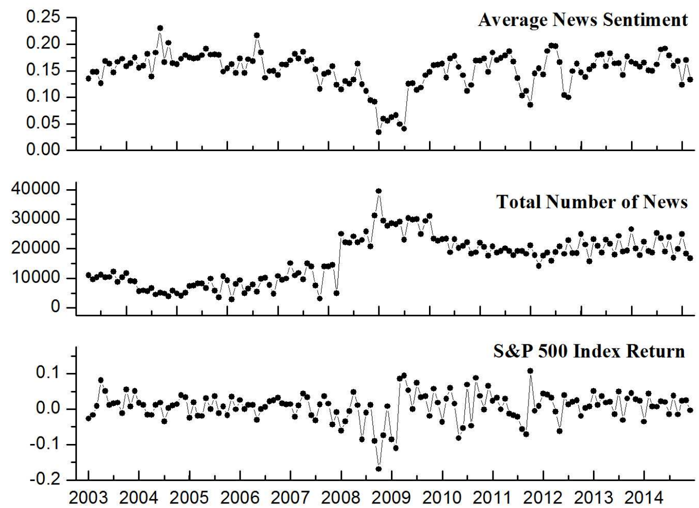


Figure 3: Monthly news sentiment data with market return. *Notes*: The figure shows time series of news sentiment, number of news, and S&P 500 return. The data frequency is monthly and ranges from 2003 to 2014.

determine that news sentiment leads market return, but there is no effect on reversed direction.

5. Experiment

5.1. Sentiment indicators parameter optimization

To determine the look-back window N in sentiment shock and trend indicators, we apply Spearman rank correlation (see Equation 9) to measure quality of predicting subsequent return ranks. Higher rank correlation means stronger prediction power. As assigning one parameter for each stock is computationally intensive, we categorize the stock universe according to GICS sectors and run an optimization for each sector. The optimization rule is to maximize Spearman rank correlation between sentiment shock (or trend) indicator scores and leading 1-week stock returns during training period. We use four years of data from 2003 to 2006 to train the look-back windows for sentiment indicators.

$$\begin{aligned} \rho &= 1 - \tfrac{6\sum d_i^2}{n(n^2-1)} \\ d_i &= x_i - y_i \end{aligned} \tag{9}$$

where x_i and y_i are rank of subsequent return and rank of sentiment indicator on training input i, and n is the training sample size.

Table 2: Optimized look-back windows for sentiment indicators

Sector Name	Sentiment Shock Look-back Window	Sentiment Trend Look-back Window
Consumer Discretionary	15	14
Information Technology	11	30
Consumer Staples	18	19
Materials	15	16
Industrials	21	18
Utilities	16	28
Health Care	10	15
Energy	25	20
Financials	11	25
Telecommunication Services	19	24

Notes: This table lists 10 financial sectors covered by the selected stock universe and their optimized look-back windows for sentiment shock and trend indicators.

5.2. Learning-to-rank algorithms parameter selection

The parameters for RankNet and ListNet algorithms include the number of iterations T, the neural network architecture ω, and the learning rate η. These two algorithms use neural network as model. The model parameters are set as following. The number of hidden layers is 1, the number of hidden nodes is 10 and the learning rate is 0.00005. The backtest data is prepared in the format as stated in the methodology section.

The number of training iterations is a critical parameter for model selection. It needs to be large enough to guarantee that the model converges to the targeted values. But training with too many iterations will be time-consuming and may result in over-fitting. There are a range of metrics for model selection, such as Normalized Discounted Cumulative Gain (NDCG) [16], Mean Average Precision (MAP) [2], and Expected Reciprocal Rang (ERR) [7]. In our experiment, we use NDCG method to select appropriate training iterations for RankNet and ListNet. NDCG measures ranking quality through the gain of an item based on its position in the result list. It is an extension of Discounted Cumulative Gain (DCG). DCG calibrates accumulated gain of the whole ranked list, and the top ranks get smaller discount and higher weights. NDCG normalizes the DCG score by the maximum possible DCG of the given query. Therefore, NDCG ranges from 0 to 1 that represents similarity of two lists of rankings. A higher consistency in rankings results in a higher score. In addition, NDCG can be applied on the top N rank positions, which is called NDCG@N, to emphasize the top rankings and reduce computation complexity (see Equation 10).

$$\begin{aligned} \text{DCG@N} &= \textstyle\sum_{i=1}^{N} \frac{2^{s_i}-1}{\log_2(i+1)} \\ \text{NDCG@N} &= \frac{\text{DCG@N}}{\text{IDCG@N}} \end{aligned} \tag{10}$$

where s_i is the score of item i, and IDCG@N is the ideal DCG of top N items.

We use the first 3 years of data (from 2006 to 2009) to determine the number of training iterations. The data is split into training and validation datasets by the ratio of 7 : 3. NDCG values are calculated according to the validation dataset, and the number of iterations with highest NDCG value is selected. We determine the number of training iterations as 150 and 1500 for RankNet and ListNet respectively.

5.3. Automatic trading system

We design four strategies (long-only strategy and long-short strategy for each learning-to-rank algorithm) based on rank predictions by RankNet and ListNet ranking algorithms.

Long-only trading strategy: As we split the stock universe by 4 labels according to the predicted score, each group contains 128 stocks which is 25% of the whole stock universe. For long-only trading strategy, we buy the top 128 stocks. The stocks in the portfolio are equally weighted.

Long-short trading strategy: For long-short strategy, long position selection is the same as long-only strategy. We buy the top 128 stocks. To select stocks for short positions, we rank potential stock performance from bottom to top. As we emphasize the top ranks in the training by using NDCG@N, we label training data from the worst return to the best return. In other words, the lowest return gets to assigned to the highest label 4, and the highest return gets the lowest label 1. This transformation allow us to put more weights on lower returns, leading to more accurate forecasting of worst performers. And then we short sell the top 128 stocks which is equivalent to the 25% stocks with the lowest predicted returns. Stock holdings in this strategy are also equally weighted.

Dynamic trading process: In the automatic trading system, the model is trained by three years of historical data and backtested during the following 1 year (see Figure 4). We roll the time window forward to train and test the next model. For example, we use the data from 2003 to 2005 to generate the first model and use 2006 data for backtesting. And then we roll forward one year for the next training and testing cycle. We update the model according to the data from 2004 to 2006 and test it on 2007. Updating the model this way allows the trading system to be adaptive to market condition changes. We predict return rankings and rebalance positions every week. For each calendar week, a new portfolio is generated on the first trading day and the portfolio is rebalanced on the closing prices of that day. Daily portfolio return is calculated with stock daily closing price. The first learning-to-rank model is trained with data from January 2003 to December 2005. We backtest the four automatic trading strategies from 2006 to 2014.

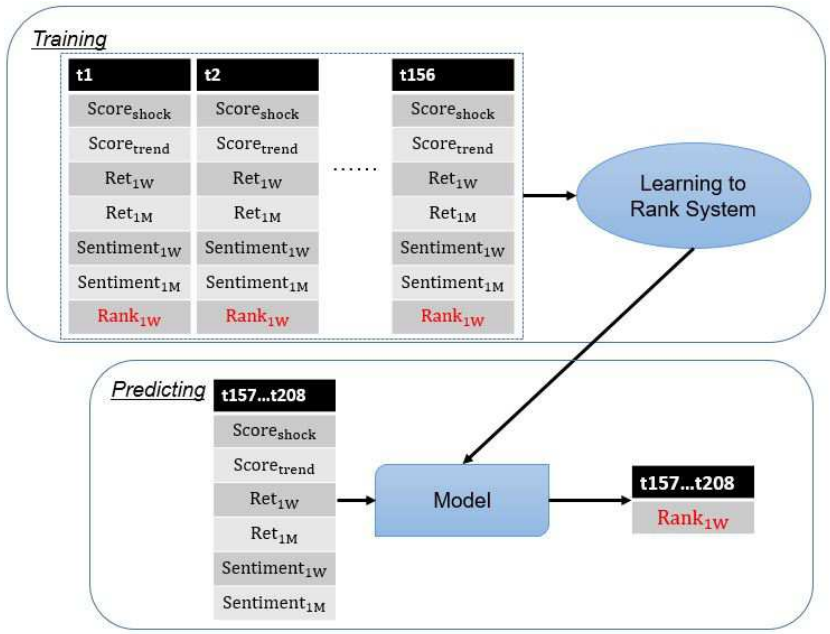


Figure 4: Rank prediction flow chart. *Notes*: This figure shows the training and predicting processes of our backtesting experiment. We apply 6 features in the model: sentiment shock and trend scores, 1-week and 1-month leading returns, and 1-week and 1-month leading sentiment.

6. Results

We run experiments of long-only and long-short strategies using ListNet and RankNet algorithms. The backtest period is from 2006 to 2014, which covers a high volatility regime during the 2008 financial crisis and the economic recovery period from 2011. To justify the performance of our strategies, we choose S&P 500 index as a benchmark for comparison.

All these strategies, especially the two using ListNet algorithm for stock selection, outperform the benchmark according to their higher cumulative returns (see Figure 5). In particular, long-short strategies present a very stable upswing even in the volatile period from 2008 to 2010. The success of long-short strategies can be justified from two perspectives. First, risk diversification among long and short positions reduces risk exposure to the market downturn during the times of financial crisis. Second and more importantly, we obtain accurate ranking forecasts based on ListNet and RankNet which lead to better returns with much lower risk. The long-only strategies show similar patterns as the S&P500 index. Their values drop in the bear market but the overall profits remain higher than the benchmark due to good stock selection.

To further evaluate each strategy in different market conditions, we split the backtest period into a high volatility regime and a low volatility regime using six-month realized market volatility. The threshold is set as two standard deviations larger than the average, which is 36.93%. The high volatility regime is from October 2008 to May 2009. During the whole backtest period, all strategies generate a higher Sharpe ratio than the S&P500 index (see Table 3). The long-short strategy based on the ListNet algorithm is the best one with 1.50 Sharpe ratio, followed by 1.07 from the long-short strategy based on the RankNet algorithm. In the high volatility regime, long-short strategies perform much better than both long-only strategies and the benchmark with a much lower maximum drawdown and volatility. In the low volatility regime, long-only strategies generate around 1.5 times higher return than the S&P500 index. Long-short strategies display a robust risk control and double the Sharpe ratio of S&P500 index. This result demonstrates that both learning-to-rank algorithms provide valuable information about rankings in subsequent returns. Moreover, the ListNet approach is more reliable in stock selection.

We also compare our results with previously proposed strategies using the same sentiment indicators and a hedge fund performance index HFRIEMN. There are a number of studies about sentiment-based trading strategies using varied sentiment computation rules [18, 26, 33]. An earlier study [31] is the most comparable work which used Thomson Reuters news sentiment and the similar sentiment indicators. Following [31], we applied sentiment trend and shock scores in a long-only trading strategy of SPY and reproduced the results using our data. Overall, the results of ListNet and RankNet long-only strategies are quite close to the results from [31] both in terms of return and Sharpe ratio, but the performances of ListNet and RankNet long-short strategies have

Table 3: Backtest results in different market conditions

	Return	**Volatility**	**Sharpe ratio**	**Max. drawdown**
ListNet long-only	15.07%	25.37%	0.59	52.90%
RankNet long-only	12.78%	25.61%	0.50	57.10%
ListNet long-short	9.56%	6.36%	1.50	10.42%
RankNet long-short	7.99%	7.49%	1.07	9.10%
Benchmark (SPY)	7.25%	21.27%	0.34	55.19%

High volatility regime

	Return	**Volatility**	**Sharpe ratio**	**Max. drawdown**
ListNet long-only	-37.47%	57.83%	-0.65	51.27%
RankNet long-only	-35.76%	57.55%	-0.62	51.12%
ListNet long-short	6.71%	13.39%	0.50	10.42%
RankNet long-short	8.26%	15.01%	0.55	9.10%
Benchmark (SPY)	-42.65%	50.53%	-0.84	47.17%

Low volatility regime

	Return	**Volatility**	**Sharpe ratio**	**Max. drawdown**
ListNet long-only	20.08%	19.66%	1.02	26.63%
RankNet long-only	17.41%	20.08%	0.87	26.83%
ListNet long-short	9.84%	5.22%	1.88	5.08%
RankNet long-short	7.97%	6.33%	1.26	7.08%
Benchmark (SPY)	12.00%	15.89%	0.76	21.49%

Notes: These tables compare the performance of the proposed trading strategies with benchmark in different market conditions. The first subtable shows return, volatility, Sharpe ratio and Maximum drawdown through the full backtesting period. The following two subtables present the same measurements under the high volatility regime and the low volatility regime respectively.

significantly higher Sharpe ratios than that of the sentiment indicator strategies without learning-to-rank approaches (see Table 4). More specifically, all the four long-only trading strategies obtain similar returns and Sharpe ratios, and they are all better than the passive S&P500 index return. Although the long-short strategies using learning-to-rank approaches produce less absolute return, the Sharpe ratio is significantly higher than all the long-only strategies, indicating their robustness under different market conditions. In terms of the maximum drawdown, the sentiment trend indicator and ListNet approach outperform the other two strategies. In addition to comparing with the passive S&P500 index return, we also want to see how the performance of our approach is compared with the general industry performance using the similar strategies. In this case, we choose the HFRIEMN index which is the Hedge Fund Research Equity Market Neutral Index, capturing the performance of long-short strategies in hedge funds. We find that the return of our long-short trading strategies almost quadruples that of the HFRIEMN index. Although our strategies are more volatile, we still obtain higher Sharpe ratio than the hedge fund index. Especially, the Sharpe ratio of the ListNet long-short strategy is about twice as much as the HFRIEMN index.

Table 4: Comparisons to other sentiment-based and technical trading strategies

	Return	**Volatility**	**Sharpe ratio**	**Max. drawdown**
Sentiment trend *	16.90%	26.50%	0.64	49.09%
Sentiment shock *	12.24%	25.22%	0.49	56.37%
ListNet long-only	15.07%	25.37%	0.59	52.90%
RankNet long-only	12.78%	25.61%	0.50	57.10%
Benchmark (SPY)	7.25%	21.27%	0.34	55.19%
ListNet long-short	9.56%	6.36%	1.50	10.42%
RankNet long-short	7.99%	7.49%	1.07	9.10%
HFRIEMN Index **	2.31%	2.88%	0.80	9.15%

Notes: The results marked with $\star$ are from a previous study [31]. The results marked with $\star\star$ are from Bloomberg.

7. Conclusion

In this paper, we design a stock portfolio selection approach utilizing learning-to-rank algorithms on two news sentiment indicators to capture relative performance of stocks. ListNet and RankNet algorithms are applied in the experiments of portfolio strategies. Our data includes features, such as historical market returns, financial news sentiment, and sentiment shock and trend scores. Our methodology combines this information to derive ranking models that facilitate stock selection. The backtest results indicate that the portfolio selected by both ranking algorithms significantly outperform the sentiment-based strategies without learning-to-rank algorithms, the market benchmark, and the hedge fund industry performance index. The superior performance is consistent under different market conditions.

The overall merit of the proposed trading strategies is that these strategies are based on investors' relative views manifested through news sentiment toward the individual stocks' future performance. The application of the learning-to-rank algorithms shows its robustness against market volatility, which is demonstrated through the consistent superior risk-adjusted returns. Furthermore, this study is based on ten years of data, and the consistent results are tested through many extreme market conditions. However, one limitation of our work is that the rebalance frequency of the trading strategies is weekly; further optimization can be done to balance the trading cost and profit.

For future research, there are several possible approaches that we will continue to pursue. In this paper, we only use financial news sentiment and previous returns as ranking features. To fully utilize the capability of the learning-to-rank algorithms, we may add additional firm performance measures as ranking features, such as fundamental indicators (e.g. P/E ratio, dividend yield and earning growth) as well as technical indicators (e.g. MACD, Bollinger Band and RSI). Another approach is to train different ranking models for stocks with different properties, since stocks may respond to indicators differently. For example, the P/E ratio has the opposite effect for value and growth stocks. Lastly, we may also try different learning-to-rank algorithms. SVM and boosting based algorithms may have better trading results than neural network based algorithms.

Reference

[1] Antweiler, W., Frank, M. Z., 2004. Is all that talk just noise? The information content of Internet stock message boards. Journal of Finance 59 (3), 1259–1294.

[2] Baeza-Yates, R., Ribeiro-Neto, B., et al., 1999. Modern information retrieval. Vol. 463. ACM press New York.

[3] Barberis, N., Shleifer, A., Vishny, R., 1998. A model of investor sentiment. Journal of Financial Economics 49 (3), 307–343.

[4] Burges, C., Renshaw, E., Deeds, M., Hullender, 2005. Learning to rank using gradient descent. Proceedings of the 22nd international conference on Machine learning - ICML '05, 89–96.

[5] Burges, C. J. C., Ragno, R., 2007. Learning to Rank with Nonsmooth Cost Functions. Machine Learning 19, 193–200.

[6] Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., Li, H., 2007. Learning to Rank : From Pairwise Approach to Listwise Approach. Proceedings of the 24th international conference on Machine learning, 129–136.

[7] Chapelle, O., Metlzer, D., Zhang, Y., Grinspan, P., 2009. Expected reciprocal rank for graded relevance. In: Proceedings of the 18th ACM conference on information and knowledge management. ACM, pp. 621–630.

[8] Crammer, K., Singer, Y., 2001. Pranking with Ranking. In: Nips. Vol. 1. pp. 641–647.

[9] Da, Z., Engelberg, J., Gao, P., 2015. The sum of all fears investor sentiment and asset prices. Review of Financial Studies 28 (1), 1–32.

[10] Dempster, M., Jones, C., 2001. A real-time adaptive trading system using genetic programming. Quantitative Finance 1 (4), 397–413.

[11] Dempster, M. A., Leemans, V., 2006. An automated fx trading system using adaptive reinforcement learning. Expert Systems with Applications 30 (3), 543–552.

[12] diBartolomeo, D., Warrick, S., 2005. Making covariance based portfolio risk models sensitive to the rate at which markets reflect new information" chapter 12 in linear factor models edited by knight, j. and satchell, s.

[13] Freund, Y., Iyer, R., Schapire, R. E., Singer, Y., 2003. An Effcient Boosting Algorithm for Combining Preferences. Journal of Machine Learning Research 4, 933–969.

[14] Healy, A. D., Lo, A. W., 2011. Managing real-time risks and returns: the thomson reuters newsscope event indices. Professor David J. Hand, Professor of Statistics, Imperial College, London; Chief Scientific Advisor, Winton Capital Management; and President, Royal Statistical Society, 73.

[15] Herbrich, R., Graepel, T., Obermayer, K., Jan 2000. Large Margin Rank Boundaries for Ordinal Regression. MIT Press, Ch. 7, pp. 115–132.

[16] Järvelin, K., Kekäläinen, J., 2002. Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems (TOIS) 20 (4), 422–446.

[17] Kaastra, I., Boyd, M., 1996. Designing a neural network for forecasting financial and economic time series. Neurocomputing 10 (3), 215–236.

[18] Khadjeh Nassirtoussi, A., Aghabozorgi, S., Wah, T. Y., Ngo, D. C. L., 2015. Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. Expert Systems with Applications 42 (1), 306–324.

[19] Kim, K.-j., 2003. Financial time series forecasting using support vector machines. Neurocomputing 55 (1), 307–319.

[20] Lee, C. M. C., Shleifer, A., Thaler, R. H., 1991. Investor Sentiment and the Closed-End Fund Puzzle. The Journal of Finance 46 (1), 75–109.

[21] Leinweber, D., Sisk, J., 2012. Relating news analytics to stock returns. John Wiley & Sons, Ltd.

[22] Liu, T.-Y., 2009. Learning to Rank for Information Retrieval. Found. Trends Inf. Retr. 3 (3), 225–331.

[23] Lu, C.-J., Lee, T.-S., Chiu, C.-C., 2009. Financial time series forecasting using independent component analysis and support vector regression. Decision Support Systems 47 (2), 115–125.

[24] Mao, H., Wang, K., Ma, R., Gao, Y., Li, Y., Chen, K., Xie, D., Zhu, W., Wang, T., Wang, H., nov 2014. An Automatic News Analysis and Opinion Sharing System for Exchange Rate Analysis. In: 2014 IEEE 11th International Conference on e-Business Engineering. IEEE, pp. 303–307.

[25] Mitra, G., Mitra, L. (Eds.), may 2011. The Handbook of News Analytics in Finance. John Wiley & Sons, Ltd., Chichester, West Sussex, UK.

[26] Mitra, L., Mitra, G., Dibartolomeo, D., 2009. Equity portfolio risk estimation using market information and sentiment. Quantitative Finance 9 (8), 887–895.

[27] Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank citation ranking: bringing order to the web. Stanford InfoLab.

[28] Robertson, S. E., 1997. Overview of the okapi projects. Journal of Documentation 53 (1), 3–7.

[29] Schumaker, R. P., Zhang, Y., Huang, C.-N., Chen, H., jun 2012. Evaluating sentiment in financial news articles. Decision Support Systems 53 (3), 458–464.

[30] Shashua, A., Levin, A., 2003. Ranking with large margin principle: Two approaches. Advances in Neural Information Processing Systems 15, 937–944.

[31] Song, Q., Liu, A., Yang, S. Y., Deane, A., Datta, K., 2015. An Extreme Firm-Specific News Sentiment Asymmetry Based Trading Strategy. 2015 IEEE Symposium Series on Computational Intelligence, 898–904.

[32] Taylor, M., Guiver, J., Robertson, S., Minka, T., 2008. SoftRank: optimizing non-smooth rank metrics. Wsdm '08, 77–86.

[33] Tetlock, P. C., 2007. Giving content to investor sentiment: The role of media in the stock market. Journal of Finance 62 (3), 1139–1168.

[34] Wang, J.-L., Chan, S.-H., 2006. Stock market trading rule discovery using two-layer bias decision tree. Expert Systems with Applications 30 (4), 605–611.

[35] Wu, Q., Burges, C. J. C., Svore, K. M., Gao, J., 2010. Adapting boosting for information retrieval measures. Information Retrieval 13 (3), 254–270.

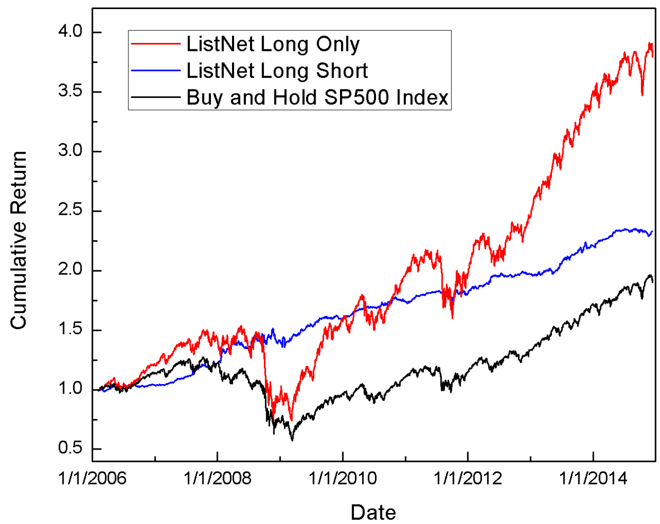
[36] Xu, J., Li, H., 2007. AdaRank: A Boosting Algorithm for Information Retrieval. Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval - SIGIR '07 49, 391–398.

[37] Yang, S. Y., Mo, S. Y. K., Liu, A., 2015. Twitter financial community sentiment and its predictive relationship to stock market movement. Quantitative Finance 15 (10), 1637–1656.

[38] Yang, S. Y., Song, Q., Yin, S., Mo, K., Datta, K., Deane, A., 2015. The Impact of Abnormal News Sentiment on Financial Markets. Journal of Business and Economics 6 (10), 1682–1694.

[39] Yue, Y., Finley, T., Radlinski, F., Joachims, T., 2007. A support vector method for optimizing average precision. Proceedings of the 30th annual

international ACM SIGIR conference on research and development in information retrieval - SIGIR '07 Amsterdam,, 271.



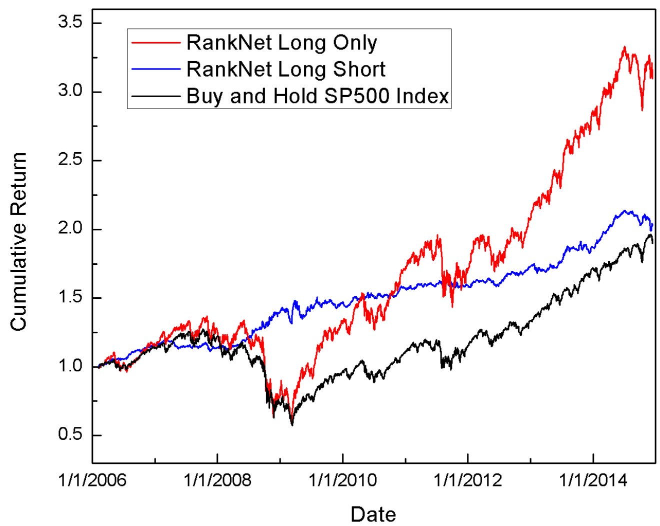


Figure 5: Portfolio cumulative return. *Notes*: This figure shows cumulative profits and losses during backtesting from 2006 to 2014. The top figure presents the long-only strategy and long-short strategy using ListNet algorithm. The bottom figure presents the long-only strategy and long-short strategy using RankNet algorithm.