Decision Support Systems 57 (2014) 212–223

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Empirical evaluation of an automated intraday stock recommendation system incorporating both market data and textual news

Tomer Geva *, Jacob Zahavi

Recanati Business School, Tel Aviv University, Israel

ARTICLE INFO

Article history:
Received 15 December 2011
Received in revised form 7 September 2013
Accepted 9 September 2013
Available online 5 October 2013

Keywords:
Stock returns
Investment decisions
Data-mining
Prediction
Recommendation

ABSTRACT

In this study we evaluate the effectiveness of augmenting numerical market data with textual-news data, using data mining methods, for forecasting stock returns in intraday trading. Integrating these two sources of data not only enriches the information available for the forecasting model, but it can potentially capture joint patterns that may not otherwise be identified when each data source is employed separately. We start with market data and then gradually add various textual data representations, going from simple representations, such as word counts, to more advanced representations involving sentiment analysis. To find the incremental value of each data representation, we build an end-to-end recommendation process including data preprocessing, modeling, validation, trade recommendations and economic evaluation. Each component of the modeling process is optimized to remove human bias and to allow us to impartially compare the results of the various models. Additionally, we experiment with several forecasting algorithms to find the one that yields the "best" results according to a variety of performance criteria. We employ data representation procedures and modeling improvements beyond those used in previous related studies. The economic evaluation of the results is conducted using a simulation procedure that inherently accounts for transaction costs and eliminates biases that have potentially affected previous related data-mining studies. This research is one of the largest-scale data-mining studies for evaluating the effectiveness of integrating market data with textual news data for the purpose of stock investment recommendations. The results of our study are promising in that they show that augmenting market data with advanced textual data representation significantly improves stock purchase decisions. Best results are achieved when the approach is implemented with a nonlinear neural network forecasting algorithm.

1. Introduction

1.1. Background

Successful strategies for stock trading decisions have clear and obvious benefits. Consequently, this topic had received much attention in data-mining studies, which report on a myriad of methods and algorithms for automated stock investment decisions.

Yet, successful construction of such models still remains a challenging undertaking. Key financial theories such as the Random Walk model of stock prices [18] and the Efficient Market Hypothesis (EMH) [19] indicate that it is difficult to generate excessive, risk-adjusted trading profits, based on currently available public information. Data-mining literature has also reported on a host of difficulties in predicting stock behavior. For instance, Dhar and Chou [11] mention that financial markets are inherently "noisy" with several types of nonlinearities. Likewise, Dhar et al. [12] refer to (a.) weak theory that results in a large number of variables, thus increasing the dimensionality of the problem; (b.) relationships between variables being weak and nonlinear; and (c.) the potential significance of variable interactions.

Most studies that predict future stock returns rely primarily on historical market data such as stock price and trading volume (e.g., [25,33,38,56]). While historical market information is relatively simple to obtain and process, the downside is that it practically ignores news concerning real-world events such as: mergers and acquisitions, dividends, interim results, analyst upgrades or downgrades, changes in management, and others—which have been documented to have an impact on stock returns [30,48].

News reports about companies abound these days and are commonly available in textual format on websites, through financial news providers such as Reuters and Bloomberg, and through other delivery systems. These data sources provide significant information about real-world events that historical market data can capture, at best, only partially and indirectly. Indeed, a host of methods, collectively called text mining or text analytics, have been developed in recent years for capturing and representing textual data. The output of such methods can potentially be used as additional explanatory information in financial prediction models. Yet, in contrast to the myriad of studies using

* Corresponding author.
E-mail address: tgeva@tau.ac.il (T. Geva).

0167-9236/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dss.2013.09.013

market-based explanatory data, relatively few studies have explored financial recommendation based only on explanatory variables obtained from textual data (e.g., [22,23,32,39,41,42,46]).

Using only textual data in prediction models also has its drawbacks, as it misses out on joint patterns of stock price changes and news publication. Examples of such joint patterns have been described in [7] and [45]. Clearly, only a trading recommendation system that is based on both market data and news data can benefit from the synergy between these two different sources of information and has the potential to detect their combined effect on stock prices.

Despite the potential benefits of integrating both types of data for predicting stock returns, this approach has been addressed in only a very limited number of data-mining studies. Among these studies we mention [50,53,54].[1] Additionally, previous data-mining studies have not conducted a systematic evaluation of the informativeness of augmenting market data with textual-news data.

In addition to being investigated in the data-mining literature, algorithmic trading has also been discussed in the financial literature. Hasbrouck and Saar [27], Hendershott et al. [28], Hendershott and Riordan [29] as well as Domowitz and Yegerman [15] examined how the stock market is influenced by algorithmic trading. These studies found that algorithmic trading generally improves various aspects of market quality such as liquidity and volatility. Other papers from the financial literature [1,9,51,52,55] have examined the informativeness of different data sources such as news and message boards which appear in textual format. For example, the objective of Antweiler and Frank [1] was to use message board postings to predict returns, and Tetlock [52] looked at whether media pessimism could be used to predict downward pressure on market price and trading volume. Our study contributes to the evaluation of the informativeness of textual news data by comparing it to the predictive capabilities of market data, and by measuring the incremental value of adding more sophisticated textual data representations. For this purpose we have constructed an "end-to-end" decision support system that first brings the different data representations to a "common ground" for comparison purposes, and then derives and evaluates actionable trading recommendations.

1.2. Research goal and methodology

Our objective was to conduct a systematic evaluation of the effectiveness of augmenting market-based data with textual news-based data for intraday stock recommendation decisions. For this purpose, we successively augmented market data with five different sets of textual data representations at different levels of complexity and sophistication, and measured the effect of each additional representation on predictive performance. We began with simple data representations, including a news item count and a "Bag of Words" representation, and gradually moved to more elaborate methods such as categorization into business events, news item sentiment scores and, finally, "calibrated sentiment" scores—a method proposed for the first time in this study.

We conducted this analysis while taking into account the *joint* effect of data representation and forecasting algorithm on intraday stock recommendation decisions. Our contention is that, due to the strong interrelationship that commonly exists between the data representation and the forecasting algorithm, this joint effect has considerable influence on the results of recommendation models. It is well known that a data representation that produces superior results with a certain data-mining algorithm might underperform when a different type of data-mining algorithm is employed. This contention is also supported by a previous pilot study [24] conducted as part of this research.

Specifically, to explore the influence of the forecasting algorithm, we ran each data setup using each of the following three data-mining algorithms, selected from three different "families" of algorithms: (a.) non-parametric feed-forward neural network; (b.) decision tree involving a genetic algorithm; and (c.) parametric linear logistic regression model.

To implement a systematic evaluation methodology, we have built an end-to-end modeling process that encompasses an entire array of tasks, including data pre-processing, feature creation, feature selection, forecasting algorithm, recommendations and economic evaluation. The various components were optimized for each data and algorithm combination in order to bring all modeling setups to a "common ground" for comparison purposes. The main modules included in this process are presented schematically in Fig. 1.

The output of the process consists of a series of trade recommendations over time. The economic value of the recommendations was evaluated by means of an elaborate simulation process that inherently accounts for the transaction cost of buying and purchasing stocks, thus avoiding biases that might have affected previous related data-mining studies.

Finally, to address the dynamics of this process over time, we used a "sliding window" approach, in which we used three months' worth of data to train a model and then validated the model's performance on data from the succeeding month. We used 11.5 months of available data in our research, resulting in 9 successive modeling setups and 8.5 months of independent data for validating the modeling performance. Fig. 2 illustrates the sliding window approach.

1.3. Intraday prediction challenge

Our study focuses on the short-term, intraday level, where stocks are purchased and sold during the same trading day. We chose to work at the intraday level since it provides a good test-bed for studies dealing with stock recommendation decisions while using news-based data. This is because utilization of intraday data is challenging due to their inherent "noisiness", and intraday trading is generally considered a risky practice.[2] On the other hand, intraday trading provides significant opportunities for automated, news-based, trading systems. These opportunities arise from the short time intervals during which stock prices may remain inefficient after news reaches the market. It is during this time that automated trading systems, reacting rapidly to new information, can potentially render profitable trades. The length of the interval in which the stock price remains inefficient has been researched in various financial studies. Studies such as [7] and [8] observed stock price inefficiency during intraday time intervals ranging from less than 1 min to 30 min.[3]

1.4. Contribution

Our main contribution is evaluating the informativeness of augmenting market data with textual-news data by means of an "end-to-end" recommendation system. This recommendation system is designed to bring different data source representations to a common ground for comparison purposes. Specific contributions include the following items, not previously reported in related data-mining literature: (a.) Using a multi-stage automated process for feature selection and transformations, designated to maximize the informativeness of market

[1] We note that [58] and [49] might have also utilized both market and textual data as explanatory variables within their forecasting models. However, the exact way in which market data are appended to textual news data is not explicitly mentioned. We also note that a few studies mentioning that they concurrently use or integrate textual news data with market data effectively utilize the market data only in relation to defining the dependent variable (e.g., [22,32]).

[2] Intraday trading usually involves substantial commissions and execution costs due to voluminous trading activity. Intraday trading usually also requires expensive data feeds as well as processing, analysis and testing software. Due to the common practice of "margin trading" in conjunction with intraday trading, risk substantially increases.

[3] [7] went further to demonstrate the feasibility of exploiting short-term market inefficiencies in the case of a financial news television program.

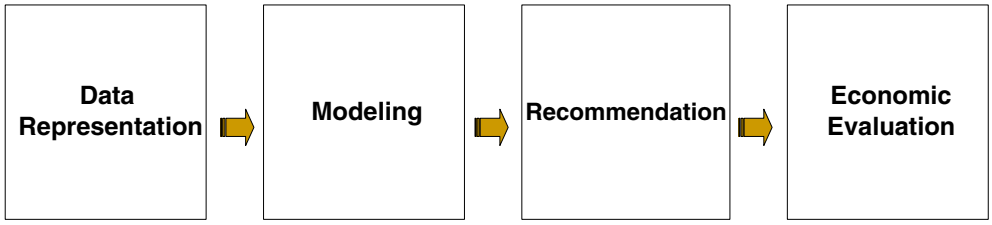

Fig. 1. Main modules.

and textual data representations. (b.) Evaluating the incremental usefulness, in terms of predictive performance, of *multiple* textual data representations, going from simple to more sophisticated representations. (c.) Devising a new sentiment analysis procedure, "sentiment calibration", designed to extract more predictive power from textual data. (d.) Experimenting with multiple forecasting algorithms alongside multiple textual data representations, to evaluate the informativeness of the different data sources and to capture the joint effect of each combination of data representation and forecasting algorithm. (f.) Using a dual-threshold optimization approach for issuing trading recommendations. (g.) Accounting for transaction costs in evaluating the economic value of the recommendation decisions.

The rest of this paper is organized as follows: Section 2 presents the raw data used in this study and the various methods employed to represent market and text-based news data. Section 3 describes the modeling module. In Section 4 we present the recommendation module responsible for issuing trading decisions, and in Section 5 we discuss the economic evaluation of the recommended trading scheme. We present and discuss the results in Section 6 and conclude this work in Section 7.

2. Data representation module

As mentioned above, one of the key purposes of this research is to evaluate the incremental value of using textual news items, on top of market data, to improve intraday stock trade recommendations. We do this using a gradual process, beginning with market data only (Section 2.3) and then incrementally augmenting the market data with five different textual data representations with different levels of sophistication (Sections 2.4–2.8). Since in data-mining it is often not known in advance which explanatory variables are more influential than others, we created a large number of potential variables for each data representation and then used an automatic feature selection process (see Section 3.1) to select the most influential variables to introduce into the model.

2.1. Raw data

Our study involves 72 S&P 500 companies during a period of 11.5 months, from September 15th, 2006 to August 31st, 2007 (see the online appendix, Section 1 for the list of companies).[4] Market data were obtained from the New York Stock Exchange (NYSE) Trades and Quotes (TAQ) database, which has been widely used in financial literature. The TAQ dataset includes two main data files: Trades and Quotes. The Trades file provides actual trading prices and trading volume originating from various market centers and exchanges. The Quotes file details the best bid prices (price suggested by a potential buyer of a certain stock) and ask prices (the price requested by the seller of a certain stock) obtained from various market centers and exchanges. Both data files are time-stamped every second.

Textual news data were obtained from the "Reuters 3000" service. This news feed also provides content from multiple other information providers (e.g., PR Newswire and Edgar Online), including financial statements, companies' announcements, and financial commentary and analysis. Overall, 51,263 news items were obtained.

In selecting the dataset for this research, we used S&P stocks that were highly traded in the market and were associated with relatively high news coverage, during a time period in which stock prices were relatively stable. We acknowledge that this dataset may not be representative of the entire stock market, though some mechanisms may be introduced in real-time trading systems to account for this factor. An example of such a mechanism, already embedded in this research, is a statistical procedure to detect cases in which our prediction process shows irregular behavior (Section 4.2).

2.2. Observations

In this study we used a time resolution of 5 min for the recommendation decisions and created one observation (or data instance) for each stock for each of these 5-minute decision points. We used the information available up to the decision point—both market data and news item data—to define the explanatory variables (as further described below), and we used the 1-h time interval succeeding the decision point as the prediction horizon for defining the dependent variable. Fig. 3 illustrates this scheme graphically.

The time resolution and the duration of the prediction horizon were selected based on an experimentation process that is described in the online appendix, Section 3. Due to the instability of early-morning trades, we began the analysis at 9:45. We stopped making predictions at 15:00 each day (1 hour before the end of the trading day). Overall, this data creation process resulted in, approximately, 1.1 M data instances.

2.3. Market data representation

Our market data representation consists of three groups of transformations: (a.) plain representation, (b.) technical analysis indicators, and (c.) piecewise linear representation.

2.3.1. Plain representation

This group of predictors consists of returns and trading volume during the current 5-minute interval and during each of the five preceding 5-minute intervals, as well as from the beginning of the trading day and from the end of the previous trading day. This group of variables also includes some simple transformations such as absolute values and binary indicators for positive/negative values. Overall, this group comprises approximately 60 predictors (see the online appendix, Section 2.2, for the complete list of market-based predictors and motivation).

[4] We chose S&P 500 companies because they include large and well-known companies. On the one hand, there is ample news coverage of these companies, but on the other hand this abundance of information is likely to cause prices to be more efficient thus prediction to be more challenging. The 11.5 months' time period and the number of companies (72) in our sample were determined by the fact that we used a semi-manual process to obtain textual news from the "Reuters 3000" service during a limited time window that we had to access the product. We used this product as a feasible alternative to purchasing a time-stamped financial news database, the cost of which would have been prohibitively high.

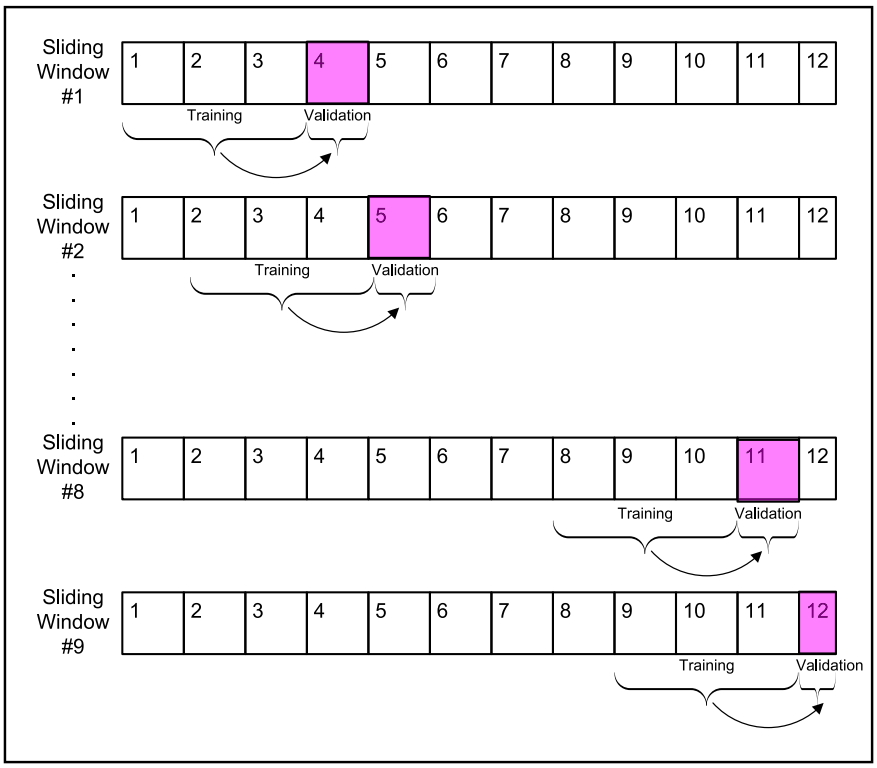

Fig. 2. Illustration of the sliding window approach.

2.3.2. Technical analysis indicators

Technical analysis is a method of investing by evaluating various statistics according to historical market activity and identifying patterns that can (potentially) have implications for future market activities. Technical analysis literature supplies a host of indicators that are reliant on various transformations of a stock's past time-series data. These indicators were employed in various related studies such as [11] and [53].

In our work we followed [53] to define the list of technical analysis indicators. However, unlike [53], which utilized inter-day data, we calculated these statistics based on intraday data, obtaining a set of 10 different indicators in all. These indicators were calculated, for each 5-minute decision point, over 3 different time durations: the preceding 10 five-minute intervals, 50 intervals and 200 intervals. (See the online appendix, Section 2.3, for the complete list of variables.)

2.3.3. Piecewise linear representation

We used the Piecewise Linear Representation (PLR) method to divide the continuous time-series data of the stock prices into linear segments, thereby obtaining a compact representation of the time series while still maintaining a good approximation of the original data.

Keogh et al. [31] describe three main PLR approaches: "bottom up", "top down", and "sliding window".[5] We used the sliding window approach (with the sum of squared errors criteria) because it is straightforward to implement and, more importantly, is suitable for rapid online calculation, as it enables the PLR segment to be determined immediately when new data points are added. For each decision point we took the last five linear segments created by the PLR algorithm and used their coefficients and durations as explanatory variables for the prediction models. See the online appendix, Section 2.4, for the complete list of PLR-based variables, and the online appendix, Section 8, for a detailed explanation of the PLR method.

[5] [31] develops an improved online algorithm.

2.4. Simple news-item count

Our first form of textual data representation involves a simple news-item count. News items were counted at various time lags, going backwards in time: the current 5-minute interval (i.e., counting the number of news items for each stock, during the last 5 min), the preceding five 5-minute time intervals, from the beginning of the trading day, and the time interval from the beginning of the current day and the end of the previous trading day. As different data providers specialize in different forms of news and content, we counted not only the news items created by Reuters but also items from external sources featured in the Reuters news feed, including: Business Wire, PR Newswire, EDGAR Online, and MarketWatch. In addition, we counted the number of news items that were identified as "breaking news" (within the Reuters data set, these items were identified by the headline format). See the online appendix, Section 2.5, for the complete list of variables based on the news-item count.

2.5. Bag of Words (BOW)

The BOW approach is a popular approach for representing textual data and has been employed in various related studies such as [20,23,32,41,42,49]. Here we used a binary indicator to represent each word within the set of news items and another binary indicator to represent each combination of two adjacent words. The BOW approach usually yields thousands or even tens of thousands of potential predictors. In order to reduce the dimensionality and increase the predictive power of the BOW, we employed several common pre-processing steps to filter the BOW variables, including stemming [44] and "stop list" filtering (including common articles and prepositions). We then used the Information Gain method to rank the set of BOW variables that had passed the filtering process. We selected as potential predictors the 400 top-ranked variables based on single words, and the top 100 predictors based on two adjacent words.

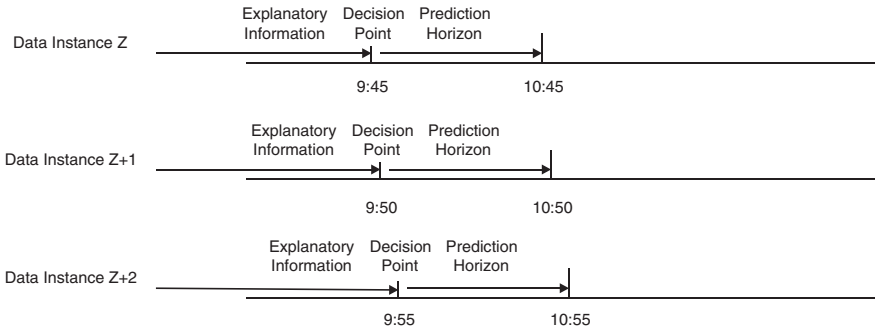

Fig. 3. Data instances for Stock X on a certain day.

2.6. Categorization into business events

While the BOW procedure produced a set of predictors based on each "important" word in the news items, the resulting set of predictors did not capture the content of the news items. To introduce the content perspective into the process, we classified each news item according to 16 different pre-defined business categories (the categories are not mutually exclusive).[6]

The categorization process was conducted automatically using supervised classifiers. For each one of the categories we tested several supervised learning-based classifiers, including: SVM, Naïve-Bayes, and J48 tree (implemented in the WEKA software [57]). For the training process, we used the Information Gain method to rank all BOW indicators (single words and pairs of adjacent words) that had passed the stemming and stop list filtering procedures, and subsequently created several alternate sets of predictors, comprising, respectively, the top-ranked 1000, 500, 100, 50, and 30 BOW predictors. We then selected the "best" classifier for each individual category, according to the F-measure of the positive class over the validation set. In addition, we used manually tuned rules, which were constructed using regular expressions and conditional statements (such as: and, or, not).

To train and validate the classification models, we used a dataset of 1500 news items that were randomly selected from the first 3 months of data, and manually assigned them tags according to the categories with which they were affiliated (each category was represented by a binary variable). After removing 63 irrelevant news items (e.g., private activities of former company executives) as well as 120 news items that reported on previous market activity for multiple companies simultaneously, we ended up with a dataset containing 1317 news items. These news items were split into a training set (2/3 of the news items), and a validation set (the remaining 1/3 of the news items). We used the "best" classification model to build the classifier for each category.

Lastly, we reintroduced the 120 news items containing repetitive information (as above) into the sample (maintaining a training/validation split of 0.666/0.334) and used this extended set to construct an additional classifier to detect news items with repetitive information (which we refer to as category 17 in our list below).

The selected classifiers for each of the 17 categories and their respective performance are detailed in the online appendix, Section 5. To create explanatory variables for our financial forecasting models, we first applied the selected classifier, for each category, to the entire set of news items (51,263 items). Subsequently, we counted the number of times each category appeared during various time intervals, as in Section 2.4. See the online appendix, Section 2.8, for a complete list of categorization-based variables.

2.7. Sentiment scores

Another factor that potentially influences market response to a news item is the sentiment, or tone, conveyed by the news item.

Sentiment analysis is concerned with analyzing texts for the sentiment that they convey, whether positive or negative. The literature describes a myriad of methods to perform sentiment analysis (see [43] for a detailed overview of this topic). The effect of sentiment in textual data on stock returns has also been previously discussed in several financial studies such as [1,9,51].

We used the commercial sentiment scoring software by Digital Trowel, Ltd. [13,14,21] to assign a sentiment score to each news item in our dataset. This software uses an advanced hybrid model that relies on both conditional random fields and rules-based methods to detect and quantify expressions of sentiments within a text.

We ran this software over all news items in our dataset, assigning each item a score between −1 (most negative) and +1 (most positive). For each decision point, we used the average sentiment as predictors. The averages were calculated for documents included in the same time intervals described in Section 2.4. See the online appendix, Section 2.9, for a complete list of sentiment-based variables.

2.8. "Calibrated" sentiment scores

We attempted to enhance the sentiment analysis process by adjusting, or "calibrating", the sentiment scores in order to reflect the extent to which we believed the information conveyed in a given news item would have a positive or negative influence on stock returns. We refer to the resulting scores as "calibrated" scores. For this purpose, we developed a supervised learning procedure, involving linear regression, to calibrate the sentiment scores, using the dataset of classified news items described above, in order to train and validate the classifiers. We first manually tagged[7] each news item based on our "judgment" regarding how the news would affect the stock price—assuming the news had not been previously published and ignoring any background knowledge about the stock.

We then used the training set (869 news items) to build a linear regression model[8] explaining the manually-assigned scores as a function of the sentiment scores obtained from the sentiment software (Section 2.7) and 16 binary variables representing the different

[6] We started with 70 categories based on categories previously reported by [39] and [53] with additional categories suggested by us. Since supervised-learning classification algorithms require at least a small number of "positive" examples, we eliminated categories with less than 20 "positive" examples in the training set, reducing the number of categories to only 16.

[7] We assigned each item a score between −1 (most negative) and +1 (most positive).

[8] The model does not utilize company specific indicators and therefore does not include company specific effects.

categories detailed in Section 2.6. Then, using the remaining 448 news items, we validated the regression results, by calculating the correlation coefficient between the news items' "calibrated" sentiment scores, produced by the linear regression model, and their manually-assigned scores. This process yielded a highly significant correlation coefficient of 0.72, with a P-value of 1.254E-75.[9]

Having validated the calibrated scores model, we applied the resulting model to the entire set of 51,263 news items to produce the corresponding calibrated sentiments. As above, for each stock, we used the average calibrated sentiments across documents in various time intervals (described in Section 2.4) as explanatory variables in our financial forecasting models. See the online appendix, Section 2.10, for a complete list of calibrated sentiment-based variables.

We note that the use of document categories as predictors was based on the reasoning that our judgment of how a news item would affect the stock price may differ according to the category, even if the corresponding news items convey a similar sentiment. For example, positive news concerning major unexpected profits is likely to have a different impact on the stock price compared with routine product announcements, which normally carry a positive tone. Indeed, the notion of a relationship between category and sentiment has been previously discussed in the literature. For instance, [17] mentions that "intuitively, the expression of sentiment in text is dependent on the topic."

Besides linear regression, we also experimented with a few other algorithms to compute calibrated sentiment scores. These algorithms included neural networks and SVM, available in the WEKA software [57]. However, at best, these algorithms improved the results only marginally. As a result, we decided to retain the simple linear regression model.

Other studies, such as [16,17,40], have used different approaches to handle topic and sentiment, and [35], for example, detected sentiment simultaneously with (latent) topics while utilizing the Latent Dirichlet Allocation (LDA) method [3]. There are several important differences between our sentiment calibration process and the approaches used in previous studies dealing with both topic and sentiment. First, our procedure is a post-process designed to adjust or calibrate existing sentiment scores rather than being a standalone process. Second, it uses a supervised learning procedure to enhance a non-supervised sentiment scoring mechanism. Third, the supervised learning is intended to replicate a human evaluator's mapping between the tone of a news item, its category, and a contextual goal. In our case, the contextual goal is rating the news item's expected effect on stock returns.

3. Modeling module

The role of modeling in data mining is to predict the value of a future event by means of a host of explanatory variables (also referred to as independent variables or predictors). As Fig. 3 above shows, our objective is to predict the probability of changes in stock returns by the end of the hour succeeding each decision point (the prediction horizon), by means of the explanatory variables generated by the data representation module described in Section 2.

To reduce prediction bias, we used a double-scoring approach to drive the recommendation decisions. This method involved building two predictive models for each dataset/algorithm combination: A "positive" model to predict the probability that stock returns will increase by more than 1% by the end of the prediction horizon (*ScrPos*), and a "negative" model to predict the probability that stock returns will decrease by more than 1% by the end of the prediction horizon (*ScrNeg*). We issued a trading signal to buy a stock only if *ScrPos* was "large" enough and *ScrNeg* was "small" enough, based on threshold values that had been predetermined according to economic considerations (see Section 4.1). This approach was chosen on the basis of a previous experiment (see online appendix, Section 3), in which the double scoring method was shown to improve the average returns and reduced variance in comparison to a single scoring method. Fig. 4 presents the double scoring design. We note that the double scoring approach is a common "bundling" method [5] used in data mining as a means of reducing prediction bias.

The modeling module includes two main components: feature selection and forecasting algorithm. Each component is described in detail below.

3.1. Feature selection

The purpose of the feature selection process is to select the subset of the most influential predictors "predicting" the decision variable out of a much larger set of potential predictors [36]. The feature selection problem is one of the most complicated challenges in building large-scale data mining models. Ignoring the feature selection problem, or conducting it in a poor manner, can have detrimental effects such as overfitting and is likely to damage overall results.

In our case, feature selection is especially important, as we incrementally add groups of text-based explanatory variables to a group of market-based explanatory variables, thus substantially increasing the number of potential predictors. Furthermore, the different data representations may contain overlapping information that could render some variables that were previously regarded as relevant—now irrelevant, and vice versa.

In this study we employed the GainSmarts software package [34], which is noted for its automatic, rule-based expert system for feature selection. In addition, GainSmarts applies a sophisticated transformation procedure to the explanatory variables to help capture nonlinear relations between the dependent and the independent variables. Specifically, multiple transformations are created for each original variable, depending upon the variable type (continuous, ordinal, nominal), each constituting a set. An example of a nonlinear transformation is representing a continuous variable as a step function using the 5 quintiles as the breaking points. All sets corresponding to the same variable constitute a cluster.

By and large, GainSmarts treats each set of predictors as a whole unit—either the entire set makes it into the model, or the entire set is out. Individual predictors in a set are eliminated only if they are highly correlated with other member variables in the set or if they cause perfect or near-perfect multicollinearity. Since all sets within a cluster are close substitutes, being different representations of the same variable, GainSmarts selects only one set in each cluster to be incorporated into the model, or none.

The process of filtering out the "weak" variables and sets is conducted by means of a rule-based expert system consisting of several levels: (a.) individual predictor analysis, (b.) intra-set analysis, and (c.) inter-set analysis. At each step of the process, GainSmarts calculates a host of uni-dimensional and multi-dimensional statistical measures. A collection of rules, using these measures, are then applied to determine the status of each variable or set, i.e., whether to eliminate the variable/set from the process or retain it for further consideration. The predictors that made it through the inter-set analysis phase constitute the input for the model building phase and for estimating the model parameters. We note that in regression-based models, the feature selection process and the model building process are intertwined. GainSmarts starts by estimating the model coefficients for the current variables that made it to the model through the inter-set analysis and eliminates the variables that do not meet the P-value threshold levels. GainSmarts then re-estimates the model parameters with the remaining predictors and further eliminates predictors that do not meet the P-value criteria and so on, until no more variables are worth eliminating from the model. The rules driving the process are predefined based on domain knowledge, experience and statistical theory. GainSmarts comes with a set

[9] We used two evaluators to rate the 448 news item set (inter-rater correlation was 0.74 with a P-value 5.06E-79). We also note that Digital Trowel's sentiment scores had a considerably better correlation (0.33, P-value of 7.4E-13) with the raters' average score than did dictionary-based methods using the financial lexicon by Loughran and McDonald [37]. The correlation between average raters' score and the dictionary-based score was 0.11 (P-value 0.01) using unweighted word counts, or 0.12 (P-value 0.006) using TF-IDF weighted word counts.

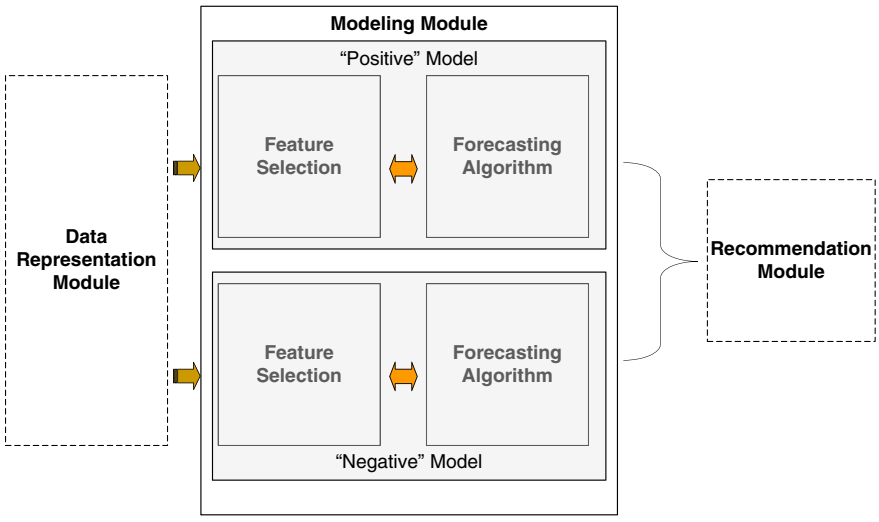

Fig. 4. Double scoring design.

of default rules that have been calibrated in an extensive experimentation process involving numerous datasets and models.

We note that by applying a systematic feature selection process to all modeling/data representation combinations, we eliminate human bias, thus bringing all modeling setups to a common ground for comparison purposes.

3.2. Forecasting algorithms

To evaluate the joint effect of data representation and forecasting algorithm on modeling performance, we run our modeling process multiple times using three different data-mining algorithms: (a.) a feed-forward neural network algorithm (Section 3.2.1); (b.) decision trees involving a genetic algorithm (Section 3.2.2); and (c.) stepwise logistic regression (Section 3.2.3).

3.2.1. Neural network (NN)

The neural network (NN) model is a well-known biologically-inspired model [47]. The NN is represented by a weighted directed graph containing three types of nodes ("processing units" or "neurons") organized in layers: the input nodes, the hidden nodes and the output nodes.

A unit (node) receives input signals from a previous layer, aggregates those signals based on an input function I, and generates an output signal based on an output function (also referred to as a transfer function), O. The output signal is then routed to the other nodes in the network as directed by the configuration of the network.

Each branch connecting any two nodes is characterized by a weight representing the strength of the connection between the nodes. These weights are determined through a training process in which the NN receives multiple examples of past cases for which the actual output is known, thereby inducing the system to adjust the strength of the weights between the nodes. Then, given a set of new observations, these weights are used to assign an NN score to each observation. Usually, a higher NN score indicates a higher likelihood of the event.

Finding the values of these weights requires solving an optimization problem. The most common method is the feed-forward algorithm, which consists of two phases: feed-forward and backwards-propagation. In the feed-forward phase, outputs are generated for each node on the basis of the current weights and are propagated to the output nodes to generate the total sum of squared deviations. Then, in the backwards-propagation phase, errors are propagated back, layer by layer, and the weights of the connections between the nodes are adjusted to minimize the total error. The feed-forward and backwards-propagation phases are executed iteratively once for each epoch, until convergence.

The strongest property of the NN network is that it inherently accounts for nonlinear relationships and complex interactions between the dependent and independent variables, making it especially applicable for nonlinear problems.

3.2.2. Decision tree involving a genetic algorithm

Decision trees are supervised learning models whose purpose is to partition a list of observations into a set of mutually-exclusive and exhaustive groups, or segments, so that all observations (e.g., customers) within a segment are "similar" (e.g., in their purchasing patterns) [6].

There are many types of decision trees. All trees evolve from a "root" node that contains all the observations in the training set. The root node is partitioned into several "children" nodes, which depend on the tree algorithm. Based upon the splitting criteria, some of the children nodes are declared as "terminal" nodes, and the others become "father" nodes to be split further in subsequent stages. This process continues until none of the nodes in the tree can be further partitioned. The results of the decision tree process are expressed by means of rules specifying which group, or segment, each observation belongs to. These rules can then be applied to a new set of observations, i.e., to assign each new observation to the appropriate segment.

In our study we used a genetic algorithm (GA) to build a tree [26]. GAs, like the NN model, are also biologically-inspired, and are based on the principle of "survival of the fittest". In decision trees, GAs serve as an engine to grow the tree and generate candidate splits for a node using a reproduction process that involves "mutations" and "cross-overs". Unlike other tree algorithms, such as CHAID, which split nodes based on one variable at a time, GAs can use any number of variables to split a node. However, the computational volume increases dramatically with the number of variables. GainSmarts allows for up to 7 variables to split a node, for computational reasons, we limited this number to 3.

3.2.3. Stepwise logistic regression (SLR)

Logistic regression models are at the forefront of discrete choice models. Most common is the binary model, where the dependent variable, Y_i, is a simple yes/no, which is coded as 0/1: 0 — for "no" (e.g., no increase in stock price), 1 — for "yes" (e.g., positive increase in stock price). Here we used the logit model and employed a stepwise procedure (hence the name Stepwise Logistic Regression), conducted as part of the inter-set analysis of GainSmarts, to select the most influential predictors in the model. In the most elaborate case, the stepwise regression involves a series of steps where at each step a variable, or a group

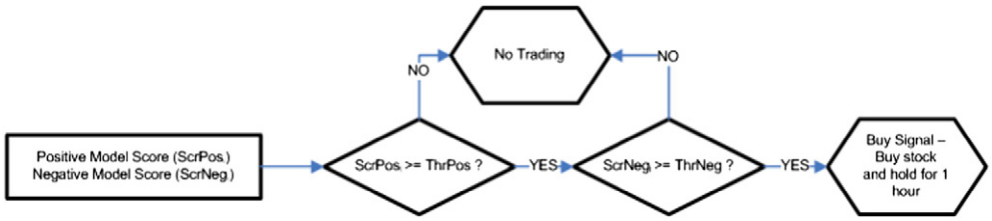

Fig. 5. Initiating a trading signal for Data Instance *i*.

of variables, is introduced into the model or eliminated from it based on F-tests. The process ends when no individual predictor or group of predictors is worth eliminating from or introducing into the model. (See online appendix, Section 10, for a detailed description of the algorithm).

4. Recommendation module

The recommendation module is responsible for issuing trading recommendations and includes two main components: (a.) threshold optimization—which is carried out offline; and (b.) initiation of trading signals and identification of conditions in which the model fails—which is executed online (i.e., during trading).

As mentioned above, the trading signals are based on a double scoring mechanism. Given the probability estimates emerging from the modeling module for each data instance *i*, we denote: (a.) $ScrPos_i$—the probability that stock returns will increase by more than 1% by the end of the hour; and (b.) $ScrNeg_i$—the probability that stock returns will decrease by more than 1% by the end of the hour. We issue a trading signal if the following condition is satisfied:

$$(ScrPos_i >= ThrPos) \text{ and } (ScrNeg_i < ThrNeg)$$

where *ThrPos* and *ThrNeg* are pre-specified threshold values for the "positive" and "negative" models, respectively. Fig. 5 graphically displays the conditions for initiating a trading signal.

Trading signals initiate a "buy" trade of stocks, which are held for a period of one hour, and then sold. In the online appendix, Section 6, we extend the recommendation engine to also consider a dual long and short trading strategy, reaching similar findings.

4.1. Threshold optimization

Selecting values for the threshold levels (*ThrPos*, *ThrNeg*) constitutes an important problem. An easy way out, one that has been taken in various studies, is to choose arbitrary values, usually 50%. However, using arbitrary cutoffs ignores economic considerations and may result in unprofitable investment, or potentially leading the investor to miss out on profitable investment opportunities. We therefore used the following optimization procedure to select the combination of threshold levels *ThrPos* and *ThrNeg* that maximizes the sum of net returns[10]:

$$MAX\left(\sum_{i \in Trans}(R_i - TransCost)\right)$$

where:

R_i is the one-hour future intraday returns.
Trans is the set of transactions satisfying the recommendation condition above.
TransCost is the transaction cost of buying and selling stocks.

For simplicity, we assumed a fixed cost per transaction, which we set at a value of 0.2%.[11] We also assumed that each buy/sell transaction involves a fixed sum.[12]

We conducted the actual optimization over the training datasets, repeating the optimization process for each new "sliding window" period. Since optimization is carried out offline, and as thresholds are determined prior to the trading period, we used a "brute force" search algorithm that experimented with various possible *ThrPos* and *ThrNeg* combinations.

4.2. Identifying conditions in which the model fails and halt trades

Predictive modeling is highly dependent on the stability of the underlying process, in our case the process governing the relationships, over time, between future price changes, historical price changes and financial news. However, as is evident from many historical and recent examples (e.g., financial turmoil in 2008 and 2011), "normal" market behavior may be interrupted by unexpected major macroeconomic events (e.g., US credit rating downgrade), abrupt changes in investors' preferences, short term events[13] and other factors.[14] These changes may occur gradually or rapidly and can eventually render a forecasting model invalid. Additionally, it is possible that certain unexpected conditions might occur only during a short period of time, after which market conditions return to "normal".

In real life, when "dramatic" changes in market conditions occur, or in cases in which the model constantly mal-performs, safety mechanisms—such as halt trades—are likely to be invoked, either manually or automatically, until "normal" conditions are restored. Related studies have reported on safety mechanisms used when executing trades (e.g., [42]). Other studies, such as [59], propose using hypothesis testing in order to detect changes in model behavior over time-series data. However, while [59] focuses on detecting changes periodically, we need to perform this detection in "real-time", in order to react to events instantaneously and reduce potential losses.

In our case, we use a test of hypothesis to identify cases in which the model is clearly mal-performing, indicating that it is appropriate to temporarily halt trades. We use a sliding window sample to test the following hypotheses:

H0. Average returns for the recommended trades are positive

H1. Average returns for the recommended trades are non-positive

A simple *t*-test, is conducted each time a trade is "completed" and the stock is sold back. We consider the preceding 100 trading signals

[10] To express *total profits* we simply multiply the expression in equation by a constant (the fixed investment sum per trade).

[11] This value is conservative in comparison to values used in other studies in this domain (e.g., [53] used a value of 0.1%, and other studies totally ignore the transaction costs). The 0.2% transaction cost is only used for the purpose of threshold optimization. In our economic evaluation of the prediction performance (Section 5) we calculate actual execution costs and brokerage fees.

[12] Various related studies have considered a fixed sum per trade, for example: [32,50].

[13] We note that it is difficult to model short term events such as the "market sellof" during April 2013 when hackers have compromised and reported a fake event at the Associated Press twitter account. See http://online.wsj.com/article/SB10001424127887323735604578440971574897016.html

[14] According to [10]: "Financial markets are at the same time recurrent and evolutionary. Old patterns repeat, albeit unpredictably, and new patterns emerge constantly".

Table 1
Returns over the validation months as a function of investment levels.

Data representation	Algorithm	$100K	$200K	$300K	$400K	$500K
Market	SLR	−1.43%	1.54%	2.03%	2.39%	2.46%
	NN	5.44%	4.58%	3.48%	2.90%	2.89%
	GA	−0.22%	−0.11%	1.72%	2.31%	2.21%
Market, simple news count	SLR	2.08%	2.59%	3.38%	3.46%*	3.23%*
	NN	5.92%	5.49%	3.92%	2.88%	2.65%
	GA	−1.34%	1.47%	2.29%	2.32%	2.21%
Market, simple news count, categories	SLR	−5.96%	−2.57%	−0.87%	0.72%	1.17%
	NN	3.72%	5.70%	5.22%	4.93%*	4.99%*
	GA	−6.09%	−1.17%	0.33%	0.06%	−0.06%
Market, simple news count, categories, sentiment	SLR	−0.29%	0.11%	1.31%	2.36%	2.96%
	NN	2.96%	7.05%*	5.50%*	4.89%*	4.85%*
	GA	−1.94%	0.55%	0.83%	1.70%	1.74%
Market, simple news count, categories, calibrated sentiment	SLR	−0.37%	−0.20%	0.86%	2.15%	2.62%
	NN	2.19%	8.57%**	6.81%**	6.35%**	6.00%**
	GA	−3.61%	0.16%	1.10%	1.56%	1.02%

Columns show returns over the validation months as a function of initial investment ($100 K–$500 K).
* *T*-test: H0: average daily returns = 0 H1: average daily returns > 0; with a P-value of 0.1 or better.Columns show returns over the validation months as a function of initial investment ($100 K–$500 K).
** *T*-test: H0: average daily returns = 0 H1: average daily returns > 0; with a P-value of 0.05 or better.

Table 2
Sharpe measure over the validation months as a function of investment levels.

Data representation	Algorithm	$100K	$200K	$300K	$400K	$500K
Market	SLR	−0.97	−0.50	−0.48	−0.41	−0.44
	NN	0.36	0.31	0.06	−0.15	−0.19
	GA	−0.83	−1.06	−0.62	−0.47	−0.62
Market, simple news count	SLR	−0.21	−0.20	0.02	0.06	−0.06
	NN	0.44	0.51	0.18	−0.16	−0.29
	GA	−1.01	−0.61	−0.43	−0.49	−0.64
Market, simple news count, categories	SLR	−1.54	−1.46	−1.35	−1.00	−0.97
	NN	0.10	0.57	0.53	0.53	0.63
	GA	−2.34	−1.55	−1.31	−1.80	−2.30
Market, simple news count, categories, sentiment	SLR	−0.63	−0.84	−0.68	−0.38	−0.16
	NN	−0.03	0.85	0.60	0.51	0.56
	GA	−1.15	−0.76	−0.90	−0.73	−0.85
Market, simple news count, categories, calibrated sentiment	SLR	−0.68	−0.94	−0.84	−0.46	−0.32
	NN	−0.17	1.19	0.96	0.97	0.96
	GA	−1.71	−1.12	−0.99	−0.90	−1.41

Columns show Sharpe ratios over the validation months as a function of initial investment ($100 K–$500 K).
For convenience, Sharpe ratios are presented as annualized values. This is done by simply multiplying the Sharpe ratio results, calculated on a daily level, by the square root of the number of trading days during one year; i.e., SQRT(250).

in order to calculate the sample mean and standard deviation used in the *t*-test.

If H0 is rejected with a P-value of 0.1 or lower, we consider this to be a signal to halt trading activities. Although at this point the recommendation module stops issuing trading recommendations, we continue to monitor all future trading signals, record their hypothetical returns (the returns that could have been obtained if the trades had been carried out), and recalculate the P-value of the *t*-test above for the preceding 100 trading signals. When the P-value is again higher than 0.1 (i.e., H0 is not rejected anymore), the recommendation module resumes making recommendations.

4.3. Extension to larger data sets

Real-life trading systems can involve much larger datasets than the dataset used in this study. Nevertheless, one trait of our system is its ability to carry out processes in parallel, thus maintaining real-time performance for much larger datasets. Instead of using a single process to monitor news and market data for thousands of companies, as well as score them to derive trading signals—one can use multiple processes or computers, each responsible for a smaller subset of companies. We expand on this topic in the online appendix, Section 7.

5. Economic evaluation

Clearly, the real test of any recommendation engine is whether it creates value. To evaluate the economic value of the trading recommendations issued by our engine, we use a simulation process that mimics the "real-life" process of buying and selling stocks. The simulation process calculates two economic measures:

1. Returns, reflecting the profitability of the trading recommendations
2. Sharpe ratio—which accounts for both returns and risk [4].

The process starts with an initial value of "available funds" for investment (AF). Each time a trade is recommended, we check whether there is still sufficient budget to carry out the transaction. If the answer is positive, we conduct the transaction and update the current AF value as well as the AF value one hour later (since our trading scheme calls for selling the stock at the end of the one-hour planning horizon). In this study we buy $5000[15] worth of stocks for each recommended trade.[16]

In addition, we assume no leverage (i.e., trades are executed only if there is sufficient unutilized budget), and no interest for uninvested funds. Finally, the Sharpe ratio and the returns are calculated based on the series of AF values at the end of each trading day.

We repeated this simulation five times using different initial levels of total funds available for investment, ranging from $100 K to $500 K, at $100 K increments. This approach helped us to capture the effect of fund availability on performance. On the one hand, when the funds allocated for investment are too small, some trading recommendations may not be fulfilled, thus harming returns and Sharpe ratios. On the other hand, if the allocated funds are too large, some of the funds could remain unutilized, also harming overall returns and Sharpe ratio.

To inherently account for execution cost, we calculated the returns of each transaction using bid and ask prices—"buying" the stock at the ask price and "selling" it at the bid price one hour later; we also subtracted from each transaction the brokerage fees as reported by a discount internet broker during this time period.[17]

The bid and ask prices were obtained from the Quotes file of the NYSE TAQ database. After filtering out irregular data using the procedures described in [2], we calculated the momentary best bid and ask prices (also known as the NBBO—National Best Bid and Offer) over all reported exchanges.[18]

To the best of our knowledge, no previous data-mining study has attempted to accurately account for transaction costs—with many studies using a fixed estimate for transaction costs, or even explicitly assuming zero transaction cost. Since most related studies achieve relatively low average returns per intraday trade (often in the range of 0.1%–0.3%, excluding transaction costs), the effect of transaction costs is significant, and may render a seemingly profitable model into an unprofitable one.

6. Results and discussion

The results obtained in this study are summarized in Tables 1 and 2, which show the total returns and the Sharpe ratio, respectively, for the various data representation methods and the three different forecasting algorithms. To avoid bias, the performance results are aggregated only over the 8.5 months of independent validation data, as exhibited in the sliding window chart (Fig. 2). That is, for each validation month,

[15] We follow related studies that used figures ranging from $1000 [50] to $10,000 [32] per (simulated) trade and settled for a fixed sum of $5000 per trade, which is near the middle of this range.
[16] In an alternate simulation reported in the online appendix, Section 4, we buy 100 stocks, reaching similar conclusions and findings.
[17] We used brokerage fees from www.interactivebrokers.com. The brokerage fees were $0.005 per share per trade (with a minimum of $1 per trade). We used fees reported for private customers with minimal level of trading. Institutional traders or traders with large trading activities received lower brokerage fees.
[18] While going beyond previous studies in this domain, we note a limitation in our method of accounting for transaction costs: TAQ quote data includes only the best bid and ask from every market center or exchange, and does not provide the full "order book" thus making it less suitable for simulating large trades.

Table 3
Returns and Sharpe measure for the best-performing algorithm.

	Returns					Sharpe				
Data representation	$100K	$200K	$300K	$400K	$500K	$100K	$200K	$300K	$400K	$500K
Market	5.44%	4.58%	3.48%	2.90%	2.89%	0.36	0.31	0.06	−0.15	−0.19
Market, news count	5.92%	5.49%	3.92%	3.46%	3.23%	0.44	0.51	0.18	0.06	−0.06
Market, news count, categories	3.72%	5.70%	5.22%	4.93%	4.99%	0.10	0.57	0.53	0.53	0.63
Market, news count, categories, sentiment	2.96%	7.05%	5.50%	4.89%	4.85%	−0.03	0.85	0.60	0.51	0.56
Market, news count, categories, calibrated sent	2.19%	8.57%	6.81%	6.35%	6.00%	−0.17	1.19	0.96	0.97	0.96

Best performing algorithm	SLR
	NN

Columns show returns and Sharpe ratios for the best performing algorithm over the validation set as a function of initial investment ($100 K–$500 K). For convenience, Sharpe ratios are presented as annualized values.

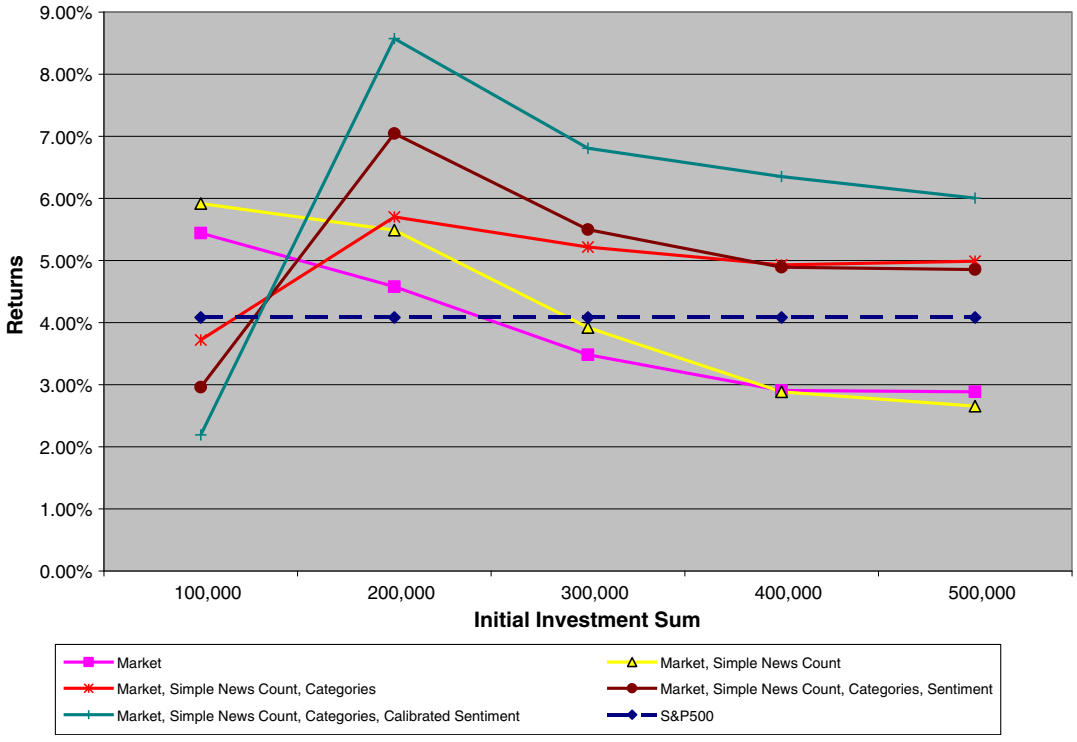

Fig. 6. Returns for benchmark and NN algorithm as a function of the initial available funds for investment.

we applied the models trained using the preceding three months, estimating the "positive" and the "negative" probabilities for each data instance and issuing the recommendation decisions. Subsequently, in calculating the performance measures, we took into account only recommendation decisions in the validation months.

In Table 3 we extract the results for the *best-performing algorithm* for both returns and Sharpe ratio measures. We observe that except for a couple of cases in which the SLR algorithm (using market and simple news-item count representations) yielded better results, the "winning" model is, undoubtedly, the NN algorithm, which produces superior results for both performance measures in almost every single case.[19]

Furthermore, as is evident from Table 1, the NN algorithm also provides the best overall results and is the only algorithm that is profitable for all data representations and initial funds available for investment.

These differences in performance can be attributed to the characteristics of the various algorithms that we used here to predict the intraday stock prices. The NN algorithm is a non-parametric learning model that has a built-in capability to capture nonlinear relationships between data elements, which often characterize financial prediction problems. In contrast, the SLR algorithm that we used in this study assumed only linear relationships between the dependent and the independent variables, whereas the GA algorithm was constrained by the tree structure and node-splitting constraints, which limited its capacity to capture the intricate relationships between the dependent and the independent variables. Incidentally, we note that the phenomenon of the NN model obtaining the best performance was also observed in a previous pilot study [24].

In Figs. 6 and 7 we display the returns and Sharpe ratio for the NN algorithm (as extracted from Tables 1 and 2) as a function of the initial investment funds and data representations and compare these results

[19] We note that in a few cases the Sharpe measures were negative, despite the fact that the overall returns were positive. This happens in cases where the model underperformed the risk-free rate (average annualized rates for 90 days U.S. T-bills was 4.7%, during the 8.5 months of validation period), resulting in a negative numerator in the formula to calculate the Sharpe ratio.

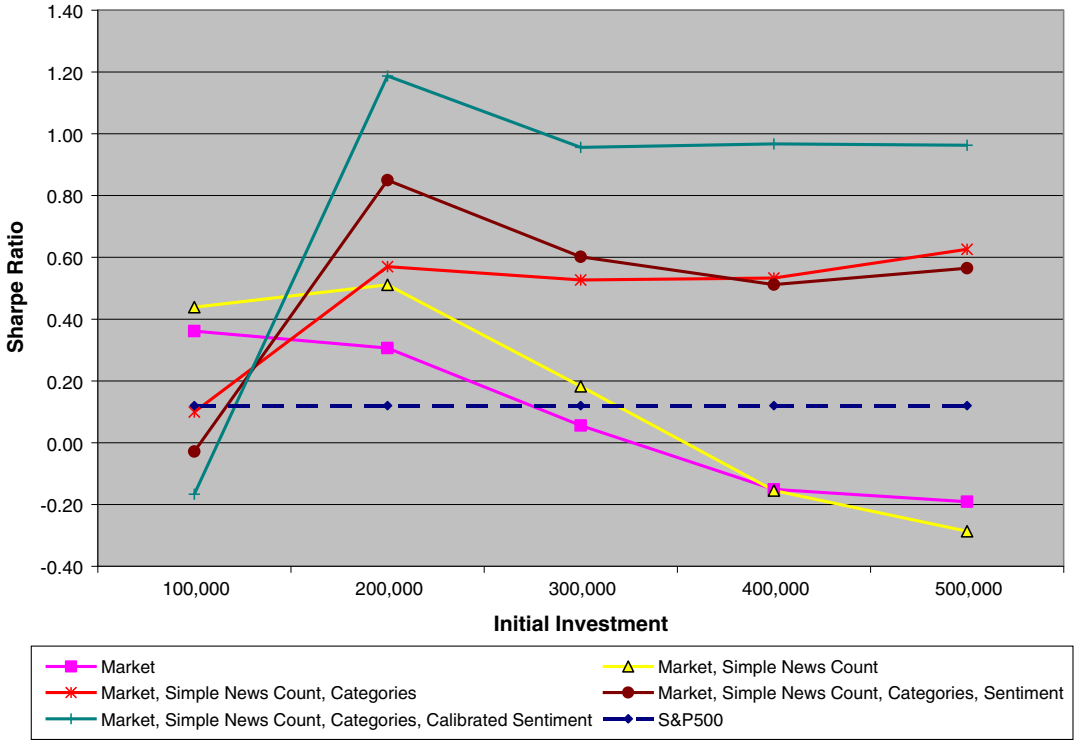

Fig. 7. Sharpe ratio (annualized values) for benchmark and NN algorithm as a function of the initial available funds for investment.

against the corresponding performance of the S&P500 index benchmark.[20] These figures clearly show that:

(a.) The NN is well equipped to utilize advanced textual data representations—the more advanced the textual data representation, the better the performance results.
(b.) When implementing the NN with the more advanced textual data representations, our model outperforms the well-known S&P 500 SPY index benchmark. The more advanced the textual data representation, the larger the gap between our modeling results and the S&P 500 benchmark. Our modeling setup also achieved statistically significant positive daily returns with a P-value better than 0.05.

The only exception to the findings above occurred in the case of initial available investment funds of $100 K—the reason being that the low investment amount was too small to fulfill all the recommendations of the model, and thus profitable investment opportunities were lost.

Overall, the bottom line is that when implemented with the "right" algorithm (NN) and the "right" data representations (incorporating the more advanced textual data representations), data-mining-driven trading recommendations certainly improve performance, as reflected in both returns and Sharpe ratios, even beyond the standard financial benchmark.

7. Conclusions

In this study we used a comprehensive modeling setup to evaluate the incremental value of various textual data representations for intraday stock trading schemes. The data module consists of several advanced data representations, some of them unique to this study, such as calibrated sentiment scores. To capture the joint effect of data representation and forecasting algorithm, we utilized three forecasting algorithms from three families of predictive models. The entire system was designed to bring the different setups to a common ground for comparison purposes. The trading recommendations were assessed by means of an economic simulation process that inherently accounted for transaction costs.

Indeed, this study shows that integrating market data with textual data contributes to improving the modeling performance and that using more advanced textual data representations further improves predictive accuracy. However, these results strongly depend on the joint selection of both data representation and forecasting algorithm. In our case, the best results were obtained with an NN algorithm using market data, simple news item counts, categorization into business events and calibrated sentiment scores as predictors. Essentially, the NN algorithm was the only algorithm that consistently managed to exploit the more advanced data representations and capture the intricate relationships between the dependent variable and the explanatory variables.

Overall, the results obtained in this study are promising enough to warrant additional research in this direction, such as evaluating the impact on performance of alternative forecasting algorithms, e.g., Support Vector Machines; evaluating the effects of additional textual data representations such as noun phrases and named entities; "bundling" the predictions of two separate models, one based solely on market data and the other on textual data; and evaluating other intraday trading policies.

Acknowledgments

The authors are grateful to Professor Ronen Feldman for the opportunity to score the set of news items with Digital Trowel's sentiment analysis software. We also wish to thank the anonymous reviewers for their valuable comments.

Appendix A

Online Appendix: http://www.tau.ac.il/~tomergev/online_appendices/Geva_Zahavi_DSS.pdf

References

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

[20] We present the results in annualized values. Similar to [1], we use the SPY fund, which tracks the S&P 500 index as a representation for this index.

[2] H. Bessembinder, Quote-based competition and trade execution costs in NYSE-listed stocks, Journal of Financial Economics 70 (2003) 385–422.
[3] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, The Journal of Machine Learning Research 3 (2003) 993–1022.
[4] Z. Bodie, A. Kane, A. Marcus, Investments, 5th ed. McGraw-Hill, New York, 2002.
[5] L. Breiman, Bagging predictors, Machine Learning 2 (1996) 123–140.
[6] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth, Belmont, CA, 1984.
[7] J.A. Busse, T.C. Green, Market efficiency in real time, Journal of Financial Economics 65 (2002) 415–437.
[8] T. Chordia, R. Roll, A. Subrahmanyam, Evidence on the speed of convergence to market efficiency, Journal of Financial Economics 76 (2005) 271–292.
[9] S.R. Das, M.Y. Chen, Yahoo! for Amazon: sentiment extraction from small talk on the Web, Management Science 53 (2007) 1375–1388.
[10] V. Dhar, Prediction in financial markets: the case for small disjuncts, ACM Transactions on Intelligent Systems and Technology 2 (2011)(paper 19).
[11] V. Dhar, D. Chou, A comparison of nonlinear methods for predicting earnings surprises and returns, IEEE Transactions on Neural Networks 12 (2001) 907–921.
[12] V. Dhar, D. Chou, F. Provost, Discovering interesting patterns for investment decision making with glower—A genetic learner overlaid with entropy reduction, Data Mining and Knowledge Discovery 4 (2000) 251–280.
[13] Digital Trowel, Semantic Identity Resolution Platform, http://www.digitaltrowel.com/solutions/WP_IR.pdf.
[14] Digital Trowel, The Stock Sonar, http://www.digitaltrowel.com/solutions/wp_the_stock_sonar.pdf.
[15] I. Domowitz, H. Yegerman, The cost of algorithmic trading: a first look at comparative performance, Trading 2005 (2005) 30–40.
[16] K. Eguchi, V. Lavrenko, Sentiment retrieval using generative models, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, The Association for Computational Linguistics, Stroudsburg, PA, 2006, pp. 345–354.
[17] K. Eguchi, C. Shah, Opinion retrieval experiments using generative models: experiments for the TREC 2006 blog track, in: E.M. Voorhees, L.P. Buckland (Eds.), NIST Special Publication 500-272: The Fifteenth Text Retrieval Conference Proceedings (TREC 2006), National Institute of Standards and Technology, Gaithersburg, MD, 2006.
[18] E.F. Fama, The behavior of stock-market prices, The Journal of Business 38 (1965) 34–105.
[19] E.F. Fama, Efficient capital markets: a review of theory and empirical work, The Journal of Finance 25 (1970) 383–417.
[20] T. Fawcett, F. Provost, Activity monitoring: noticing interesting changes in behavior, Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, 1999, pp. 53–62.
[21] R. Feldman, B. Rosenfeld, R. Bar-Haim, M. Fresko, The stock sonar — Sentiment analysis of stocks based on a hybrid approach, Proceedings of the Twenty-Third IAAI Conference, Association for the Advancement of Artificial Intelligence, Palo Alto, CA, 2011, pp. 1642–1647.
[22] G.P.C. Fung, J.X. Yu, L. Wai, Stock prediction: integrating text mining approach using real-time news, Proceedings of the IEEE International Conference on Computational Intelligence for Financial Engineering, The Institute of Electrical and Electronics Engineers, Hong Kong, 2003, pp. 395–402.
[23] G.P.C. Fung, J.X. Yu, H. Luz, The predicting power of textual information on financial markets, IEEE Intelligent Informatics Bulletin 5 (2005) 1–10.
[24] T. Geva, J. Zahavi, Predicting intraday stock returns by integrating market data and financial news reports, MCIS 2010 Proceedings, Association for Information Systems, Tel Aviv, 2010, (paper 39).
[25] C.L. Giles, S. Lawrence, A.C. Tsoi, Noisy time series prediction using recurrent neural networks and grammatical inference, Machine Learning 44 (2001) 161–183.
[26] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, New York, 1989.
[27] J. Hasbrouck, G. Saar, Low-latency trading, Journal of Financial Markets (2013) (Published, online 22 May).
[28] T. Hendershott, C.M. Jones, A.J. Menkveld, Does algorithmic trading improve liquidity? The Journal of Finance 66 (2011) 1–33.
[29] T. Hendershott, R. Riordan, Algorithmic trading and information, Working Paper, University of California, Berkeley, 2009.
[30] P.S. Kalev, W.M. Liu, P.K. Pham, E. Jarnecic, Public information arrival and volatility of intraday stock returns, Journal of Banking & Finance 28 (2003) 1441–1467.
[31] E. Keogh, S. Chu, D. Hart, M.A. Pazzani, An online algorithm for segmenting time series, Proceedings of IEEE International Conference on Data Mining, The Institute of Electrical and Electronics Engineers, Los Alamitos, CA, 2001, pp. 289–296.
[32] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, J. Allan, Mining of concurrent text and time series, Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Workshop on Text Mining, ACM, New York, 2000, pp. 37–44.
[33] W. Leigh, R. Purvis, J.M. Ragusa, Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support, Decision Support Systems 32 (2002) 361–377.
[34] N. Levin, J. Zahavi, Case studies: commercial, multiple mining tasks systems: gainsmarts, in: W. Klösgen, J.M. Zytkow (Eds.), Handbook of Data Mining and Knowledge Discovery, Oxford University Press, New York, 2002, pp. 601–609.
[35] C. Lin, Y. He, Joint sentiment/topic model for sentiment analysis, Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM, New York, 2009, pp. 375–384.
[36] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, Norwell, MA, 1998.
[37] T. Loughran, B. McDonald, When is a liability not a liability? Textual analysis, dictionaries, and 10Ks, The Journal of Finance 66 (2011) 35–65.
[38] C.J. Lu, T.S. Lee, C.C. Chiu, Financial time series forecasting using independent component analysis and support vector regression, Decision Support Systems 47 (2009) 115–125.
[39] S.A. Macskassy, H. Hirsh, F.J. Provost, R. Sankaranarayanan, V. Dhar, Intelligent information triage, Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, 2001, pp. 318–326.
[40] Q. Mei, X. Ling, M. Wondra, H. Su, C. Zhai, Topic sentiment mixture: modeling facets and opinions in weblogs, Proceedings of the 16th International Conference on World Wide Web, ACM, New York, 2007, pp. 171–180.
[41] M.A. Mittermayer, Forecasting intraday stock price trends with text mining techniques, Proceedings, Annual Hawaii International Conference on System Sciences, IEEE Computer Society, Washington, DC, 2004, p. 64.
[42] M.A. Mittermayer, G.F. Knolmayer, NewsCATS: a news categorization and trading system, Proceedings of the Sixth International Conference on Data Mining, IEEE Computer Society, Washington, DC, 2006, pp. 1002–1007.
[43] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2 (2008) 1–135.
[44] M.F. Porter, An algorithm for suffix stripping, Program: Electronic Library and Information Systems 14 (1980) 130–137.
[45] R.J. Rendleman, C.P. Jones, H.A. Latan, Empirical anomalies based on unexpected earnings and the importance of risk adjustments, Journal of Financial Economics 10 (1982) 269–287.
[46] C. Robertson, S. Geva, R.C. Wolff, Can the content of public news be used to forecast abnormal stock market behaviour? Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), IEEE Computer Society, Washington, DC, 2007, pp. 637–642.
[47] D.E. Rumelhart, J.L. McClelland, R.J. Williams, Learning internal representation by error propagation, Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, Cambridge, MA, 1986, pp. 318–362.
[48] P. Ryan, R.J. Taffler, Are economically significant stock returns and trading volumes driven by firm specific news releases? Journal of Business Finance & Accounting 31 (2004) 49–82.
[49] R.P. Schumaker, H. Chen, Textual analysis of stock market prediction using financial news articles, Proceedings of the 12th Americas Conference on Information Systems, Association for Information Systems, Acapulco, Mexico, 2006.
[50] R.P. Schumaker, H. Chen, Evaluating a news-aware quantitative trader: the effect of momentum and contrarian stock selection strategies, Journal of the American Society for Information Science and Technology 59 (2008) 247–255.
[51] P.C. Tetlock, M. Saar Tsechansky, S. Macskassy, More than words: quantifying language to measure firms' fundamentals, The Journal of Finance 63 (2008) 1437–1467.
[52] P.C. Tetlock, Giving content to investor sentiment: the role of media in the stock market, The Journal of Finance 62 (2007) 1139–1168.
[53] J.D. Thomas, News and Trading Rules, PhD Thesis: Carnegie Mellon University, 2003.
[54] J.D. Thomas, K. Sycara, Integrating genetic algorithms and text learning for financial prediction, in: A.A. Freitas, W. Hart, N. Krasnogor, J. Smith (Eds.), Data Mining with Evolutionary Algorithms, Association for the Advancement of Artificial Intelligence, Palo Alto, CA, 2000, pp. 72–75.
[55] R. Tumarkin, R.F. Whitelaw, News or noise? Internet postings and stock prices, Financial Analysts Journal 57 (2001).
[56] H. White, Economic prediction using neural networks: the case of IBM Dailystock returns, Proceedings of the IEEE International Conference on Neural Networks, vol. 2, Institute of Electrical and Electronics Engineers, San Diego, CA, 1988, pp. 451–458.
[57] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, Burlington, MA, 2005.
[58] B. Wuthrich, V. Cho, S. Leung, D. Permunetilleke, K. Sankaran, J. Zhang, Daily stock market forecast from textual Web data, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, IEEE SMC Society, New York, 1998, pp. 2720–2725.
[59] G. Zeira, O. Maimon, M. Last, L. Rokach, Change detection in classification models induced from time series data, in: M. Last, A. Kandel, H. Bunke (Eds.), Data Mining in Time Series Databases, World Scientific, Singapore, 2004, pp. 101–125.

Tomer Geva is a faculty member at the Recanati Business School, Tel Aviv University. Previously, he was a visiting scholar at NYU Stern School of Business and a post-doctoral Research Scientist at Google. Tomer holds a Ph.D. in the fields of information systems and data-mining from Tel-Aviv University. Prior to his Ph.D. studies, he gained extensive practical experience in information systems development and design. Tomer also holds an MBA degree (cum laude) in information systems from Tel-Aviv University and a B.Sc. Degree (cum laude) in Industrial Engineering from the Technion — Israel Institute of Technology. Tomer's research interests focus on understanding the utility and informativeness of large scale data for the purpose of deriving business decisions in various domains.

Jacob Zahavi is a Professor Emeritus at the Recanati Business School, Tel Aviv University in Israel, and a frequent visitor to the Wharton School and other US universities. His main areas of research and interest are data-mining and KDD (Knowledge Discovery in Databases) with more than 20 years of experience in modeling and applications. He is the mastermind behind the development of GainSmarts, a data-mining software package, which is a two-time winner of the prestigious gold miner award in the KDD CUP competition. He published about 50 papers in academic journals, several of which won academic awards for excellence. He is also involved in numerous data-mining applications in diverse fields such as the financial, automotive, retail, hospitality, and communication industries. Dr. Zahavi holds a Ph.D. in systems engineering from the University of Pennsylvania, a M.Sc. in operations research from the Technion — Israel Institute of Technology and a B.Sc. in economics and statistics from the Hebrew University, Jerusalem.