Neural Networks for Sentiment Detection in Financial Text

Caslav Bozic* and Detlef Seese*

With a rise of algorithmic trading volume in recent years, the need for automatic analysis of financial news emerged. We propose system for quantifying text sentiment based on Neural Networks predictor. Using methodology from empirical finance we prove statistically significant relation between text sentiment of published news and future daily returns.

JEL Codes: C45, D83, and G17

1. Introduction

News makes a very important information source for traders. News stories reach huge number of people, and they can initiate massive market movements, like panic selling, or massive buying, but they can also lead to more subtle market movements. Until recently it was mainly the task of human analysts to determine how positive or negative a news story is for a subject company. In general, we call such a positivity or negativity measure `text sentiment'.

With a rise of algorithmic trading volume in recent years, the need for quantifying qualitative information in textual news and incorporating that additional information in new trading algorithms emerged. This task has to be done on vast amount of data and in millisecond frequency range, so these requirements render human analysts less useful and machines have to take over the task of quantifying text sentiment.

In past decade about dozen of systems and methods appeared in the literature that try to solve this task. They use text mining of publicly accessible financial texts in order to predict market movements. They employ different machine learning approaches, define important features in text in a different way, and use different and often incomparable criteria for performance measurement. In his work (Tetlock, 2007) used fairly simple text sentiment measurement - number of words classified as ‘negative’ in Harvard IV-4 dictionary. He evaluated the results by building the regression between this text sentiment measure and future stock prices of the subject company. This proved statistically significant predicting power of the text sentiment measure. We will use this basic idea of the regression as a performance assessment tool.

---

* Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology (KIT), Germany, e-mail: {bozic, detlef.seese}@kit.edu.
Financial support from the Graduate School 895 “Information Management and Market Engineering” funded by Deutsche Forschungsgemeinschaft (DFG) is gratefully acknowledged.

2. Literature Review

In the corpus of research on influence of news on market reactions, only a humble fraction employs artificial neural networks. In their survey (Mittermayer and Knolmayer, 2006) included only one work that uses neural networks for classification - (Wüthrich et al, 1998). In this work authors propose a system that classifies news articles published on web portals during the night. Up, down, and steady are three categories that are defined depending on the influence news have on five equity indices: Dow Jones, Nikkei, FTSE, Hang Seng, and Straits Times. The goal was to forecast the trend of index daily value one day ahead. Underlying dictionary is hand-crafted, and classifiers used are naïve Bayes, nearest neighbour, and a neural network with 423 input nodes, 211 hidden nodes, and three output nodes. The results reported for the neural network classifier are somewhat worse than those reported for nearest neighbour classification.

Two works of interest that tackle market response to news, but were not included in the survey, are (Liang, 2005) and (Liang and Chen, 2005). The first work uses only volume of posted internet stock news to train neural network and predict changes in stock prices, so we can not consider the system proposed there as a real text-mining system. As an extension, the second work (Liang and Chen, 2005) employs natural language processing techniques and hand-crafted dictionary to predict stock returns. They use feedforward neural network with five neurons in the input layer, 27 in the hidden layer, and one output neuron. Since only 500 news items was used for the analysis, no statistical significance of the results could be found.

A kind of explanation for such small number of systems employing neural networks in financial text mining we might find in viewpoints similar to this one:

> Both our own pre-tests (not shown here) and comparative empirical studies provide evidence that the classification performance of SVM is superior to both parametric data mining techniques, e.g. Naïve Bayes, and non-parametric data mining techniques, e.g. k-Nearest Neighbour or Neural Networks. Moreover, as already stated above, SVM “is usually less vulnerable to the over-fitting problem [and] the solution of SVM is always unique and globally optimal”. That is the reason why we decided SVM to be the method of choice in this paper.
>
> (Groth and Muntermann, 2010)

The authors quote (Joachims, 1998) and (Yang and Liu, 1999) that compared different approaches to text classification, but forget that in recent years we witnessed the advance in neural network methodology, like fast training algorithm for deep multilayer neural networks developed by (Hinton and Salakhutdinov, 2006). The ability of neural networks to capture very complex patterns, and the new learning algorithms that enable training in acceptable time, call for reconsideration of previous statements.

3. Methodology

The proposed system is based on Neural Network predictor. The Neural Network has to be trained first. After training step the system is ready to take a text of the news story about a particular company as an input, and it produces a numerical text sentiment measure as an output. We will show that produced text sentiment corresponds with future returns of the company's stock.

As a source of financial news we use the archive of all news items published via Reuters NewsScope in year 2003. Besides news text this dataset offers additional metadata. Most important for us are the publication timestamp and the identifiers of all the companies mentioned in the news. We form a subset of all news available in the archive by choosing only the news items related to companies that are constituents of the Russell 3000 index. The Russell 3000 Index consists of the largest 3000 U.S. companies representing approximately 98\% of the investable U.S. equity market.

As a source of trading data we have an access to Thomson Reuters Tick History database. We extract opening and closing prices for all trading days in 2003 for each company from Russell 3000 index. The opening and closing prices are adjusted for dividends and then transformed into log-returns. In this way we get open-to-close (R_{OC}), open-to-open (R_{OO}), close-to-open (R_{CO}), and close-to-close (R_{CC}) returns for each trading day in 2003 and each Russell 3000 company. The respective equations are given below, where P_O and P_C represent opening and closing stock price, respectively, and t represents current trading day.

$$R_{OC} = \ln \frac{P_O(t)}{P_C(t-1)}$$

$$R_{OO} = \ln \frac{P_O(t)}{P_O(t-1)}$$

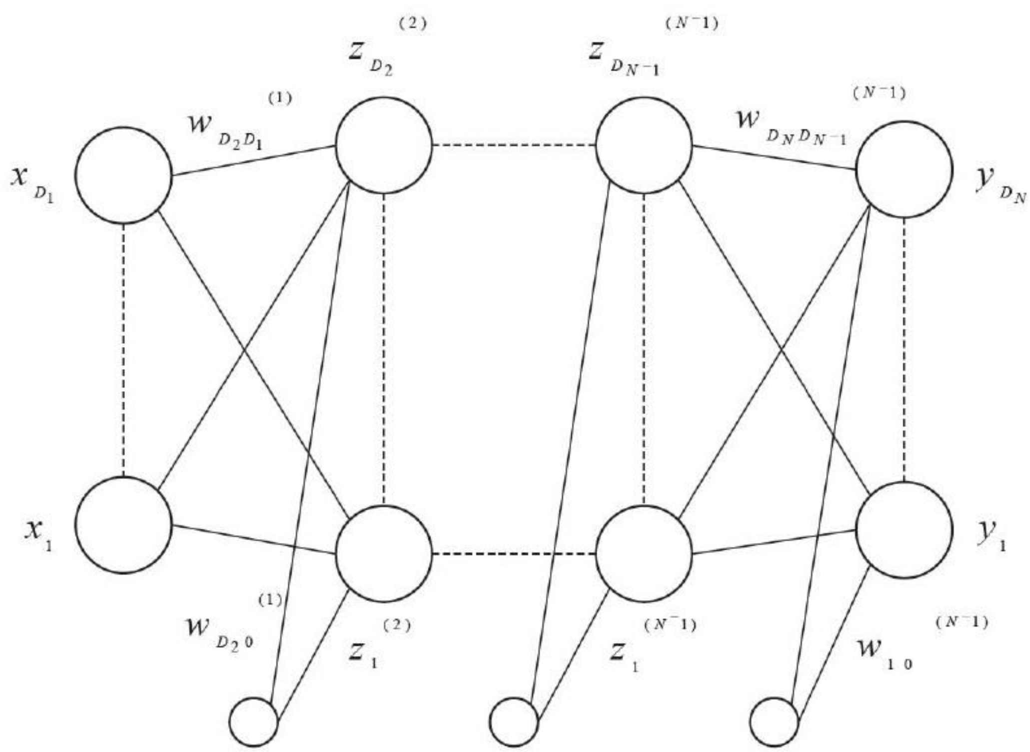
$$R_{CO} = \ln \frac{P_C(t)}{P_O(t)}$$

$$R_{CC} = \ln \frac{P_C(t)}{P_C(t-1)}$$

For training we singled out news about four companies: Apple Computer Inc., International Business Machines Corp., Microsoft Corp., and Oracle Corp. As news items can be rather long, we kept only the paragraph where the subject company is mentioned and four surrounding paragraphs. The words with only one or two characters are discarded. All other words are stemmed, and their absolute frequencies in the text are calculated. Each of 11781 distinct words in our training set represents one dimension of the training vector. Each news item represents one training vector. The target value is determined according to the next day's open-to-open return of the subject company. To decrease the overlapping between time range of news publication and time range of returns, all the news items published after the closing time of the market (3:30 pm, local time) are considered to belong already to the next date.

The system uses fairly simple Feedforward Neural Network with an input layer, two hidden layers, and an output layer. The information in Feedforward Neural Network flow only in one direction and their graph representation doesn't have any cycles. The size of input layer depends on the properties of input text and it is defined by the number of distinct words in training dataset. In our case the number of neurons in the input layer is 11781. The two hidden layers consist of 16 and 8 neurons, while output layer has one or two neurons, depending on a version of neural network, as explained below.

Figure 1: Neural network structure



The general structure of feed forward neural network is shown in Figure 1. The neural network function can be described using following equations:

$$z_{j_1}^{(1)} = x_{j_1}$$

$$z_{j_{k+1}}^{(k+1)} = h\left(\sum_{i=1}^{D_k} w_{j_{k+1}i}^{(k)} z_i^{(k)} + w_{j_{k+1}0}^{(k)}\right)$$

$$y_{jN} = z_{jN}^{(N)}$$

$$k = 1, \ldots, (N-1) \quad N \in \mathbb{N}$$

$$j_p = 1, \ldots, D_p$$

$$D_q \in \mathbb{N} \quad q = 1, \ldots, N$$

Input and output vectors are represented with **x** and **y**, respectively. Parameters ω_{ji} are weights, while ω_{j0} is denoted as *bias*. To determine the value of output, each neuron transforms its sum of inputs using function h() which is called *activation function*. In our case the activation function is logarithmic function.

We compared three versions of neural network. They differ in structure and training procedure.

- Version 1 has one neuron in the output layer, and the output value of that neuron is the value of future return.
- Version 2 has two neurons in the output layer. During the training, the neurons can take only one of two values --- zero or one. The first neuron is set to one if the future return is positive; the second neuron is set to one if the future return is negative.
- Version 3 has also two neurons in the output layer. If the future return is positive, first neuron is set to the value of that return, while the second is set to zero; if the future return is negative, second neuron is set to absolute value of that return, while the first is set to zero.

4. Findings

Each of 107266 news items published in the year 2003 that mentions any of the Russell 3000 companies is classified. The paragraphs with the mention of the subject company and the four surrounding paragraphs are singled out, the words with only one or two letters are discarded, and the other words stemmed. This word vector is fed to the Neural Network predictor, and as an output we get the text sentiment.

All the text sentiment results for one company and one day (in this case the next day starts already with closing the market – 3:30 pm local time) are averaged, and aligned with the corresponding return for the same company and same date.

$$R_{OO}(t,c) = \alpha_0 S(t,c) + \alpha_1 S(t-1,c) + \alpha_2 S(t-2,c) + \alpha_3 S(t-3,c) + \sum_{i=2}^{10} \beta_i dd_i(c) + \gamma \quad (1)$$

At this point we need the way to determine predicting power of the text sentiment measure. If the observed text sentiment measure actually correlates with the future stock returns, and if we represent the current day's return as a regression of previous sentiments (as in Equation 1), then the coefficients in front of the text sentiment measures should be significantly different from zero. We estimate regression parameters for linear regression with open-to-open return R_{OO} as a dependent variable using ordinary least squares method. As independent variables we use contemporaneous text sentiment value S(t), text sentiment value from day before S(t-1), two days before S(t-2), and three days before S(t-3). This is done with respect to the subject company c, which is represented as an additional parameter in the equation, besides time t. We order all the companies in our dataset according to their market capitalization (total market value of all shares of the company), and divide them into 10 equally sized groups. In this way we get the values for ten additional “dummy” variables dd_1 to dd_{10} (being 1 if the subject company falls into the respective group, and 0 otherwise). We include them into the regression to account for the variations of returns as a result of company's size.

The results are presented in Table 2 and Table 3. In columns we have results of three different versions of neural network, each of them represented with the best and the worst result according to statistical significance. The last column gives the values of the same benchmark applied to sentiment data produced by Reuters NewsScope Sentiment Engine (RNSE).

The coefficients in the table are estimations of following parameters from Equation 1: $\alpha_0 \ldots \alpha_3$ for lagged *sent* variables, $\beta_2 \ldots \beta_{10}$ for variables *dd2-dd10*, and γ for *Constant* factor. The statistical significance is expressed according to the Table 1. Given the observed data, p value represents the probability that null hypothesis is true. In our case, the null hypothesis is that particular coefficient is zero, hence the daily return doesn't depend on observed variable, or in other words that observed variable doesn't predict daily return.

Table 1: Statistical significance of the results

	p value
$***$	$< 1\%$
$**$	$< 5\%$
$*$	$< 10\%$
otherwise	$>= 10\%$

The performance of this type of neural network is strongly dependent on initial values of weights, which are in this case randomly assigned. This influences the instability of performance and different results between training sessions. It is represented by big difference between best and worst result of the same network. Having a benchmark at hand, we can solve this problem in evolutionary manner, simply by discarding training sessions with unsatisfactory performance.

Table 2: Coefficients of OLS regression for open-to-open returns

	Version 1 (worst)	Version 1 (best)	Version 2 (worst)	Version 2 (best)
sent	$8.92943e{-}005$	-0.00376926^{**}	0.000109779	-0.000219209
sent(-1)	0.000661898	0.00136879^{*}	0.000836959	0.000896365
sent(-2)	-0.000249269	-0.000636033^{***}	-0.000379369^{*}	-0.000406811^{**}
sent(-3)	-0.000466296	0.00107841^{***}	$8.91539e{-}005$	$-8.00234e{-}005$
dd2	0.000180383^{***}	0.000182595^{***}	0.000182081^{***}	0.000180861^{***}
dd3	-0.000148618^{***}	-0.000145703^{***}	-0.000147982^{***}	-0.000148486^{***}
dd4	-0.000103311^{***}	$-9.7867e{-}005^{***}$	-0.000104049^{***}	-0.000103594^{***}
dd5	-0.000378412^{***}	-0.000371808^{***}	-0.000378261^{***}	-0.000378552^{***}
dd6	-0.000107427^{***}	$-9.9013e{-}005^{***}$	-0.000107502^{***}	-0.000107807^{***}
dd7	-0.000471887^{***}	-0.000461132^{***}	-0.000475145^{***}	-0.000473357^{***}
dd8	-0.000628455^{***}	-0.000612976^{***}	-0.000628858^{***}	-0.000629213^{***}
dd9	-0.000700886^{***}	-0.000684040^{***}	-0.000699481^{***}	-0.000701054^{***}
dd10	-0.000787443^{***}	-0.000746639^{***}	-0.000805341^{***}	-0.000794924^{***}
Constant	0.00200064^{***}	0.00200466^{***}	0.00200009^{***}	0.00200032^{***}

Table 3: Coefficients of OLS regression for open-to-open returns (continued)

	Version 3 (worst)	Version 3 (best)	RNSE Data
sent	0.00784174*	0.0137568**	0.0103890***
sent(-1)	0.000382601	−0.00866401***	0.00218671**
sent(-2)	0.000472875	0.00396125***	−0.000937309**
sent(-3)	0.000213393	−0.00176992	−0.000879300**
dd2	0.000186241***	0.000181448***	0.000182762***
dd3	−0.000139854***	−0.000143045***	−0.000138983***
dd4	$-9.3916e{-}005^{***}$	$-9.4310e{-}005^{***}$	$-8.7328e{-}005^{***}$
dd5	−0.000358984***	−0.000363728***	−0.000371423***
dd6	$-8.0307e{-}005^{***}$	$-8.6828e{-}005^{***}$	$-8.1876e{-}005^{***}$
dd7	−0.000439835***	−0.000445554***	−0.000452748***
dd8	−0.000581378***	−0.000587837***	−0.000569948***
dd9	−0.000636196***	−0.000641807***	−0.000669534***
dd10	−0.000720365***	−0.000598841***	−0.000665957***
Constant	0.00201147***	0.00200834***	0.00200346***

5. Conclusion

We presented a system for automatic financial news analytics by determining text sentiment using Neural Network predictor. The employed machine learning method uses feedforward Neural Network with two hidden layers. The performance is assessed by empirical finance approach, which offers a possibility to prove statistical significance of the results.

From the presented results can be seen that, if measured by benchmark we proposed, some of the neural network structures can achieve performance that is comparable to state of the art systems. Future work would be extending these results by using Deep Multilayer Neural Networks with more than two hidden layers for determining text sentiment.

References:

Bozic, C. 2009, “FINDS - Integrative services”, *Computer Systems and Applications, IEEE/ACS International Conference*, pp. 61-62

Groth, S. and Muntermann, J. 2010, *Discovering Intraday Market Risk Exposures in Unstructured Data Sources: The Case of Corporate Disclosures*, pp.1-10.

Hinton, G. and Salakhutdinov, R. 2006, “Reducing the dimensionality of data with neural networks”, *Science*, 313(5786), p. 504.

Joachims, T. 1998 *Text categorization with Support Vector Machines: Learning with many relevant features*, pp. 137-142.

Liang, X. 2005, *Impacts of Internet Stock News on Stock Markets Based on Neural Networks*, p. 811.

Liang, X. and Chen, R.-C. 2005*, Mining Stock News in Cyberworid Based on Natural Language Processing and Neural Networks*, pp. 893-898.

Mittermayer, M. and Knolmayer, G. 2006, *Text mining systems for market response to news: A survey*.

Pfrommer, J., Hubschneider, C., and Wenzel S. 2010, *Sentiment Analysis on Stock News using Historical Data and Machine Learning Algorithms*

Tetlock, P. 2007 “Giving Content to Investor Sentiment: The Role of Media in the Stock Market”, *Journal of Finance*, 62(3).

Wüthrich, B., Permunetilleke, D., Leung, S., Cho, V., Zhang, J., and Lam W. 1998, *Daily prediction of major stock indices from textual www data* .

Yang, Y. and Liu X. 1999, *A re-examination of text categorization methods*, pp. 42-49.