NOVA

University of Newcastle Research Online

nova.newcastle.edu.au

Bozic, Caslav; Chalup, Stephan; Detlef, Seese 'Application of intelligent systems for news analytics', Financial Decision Making Using Computational Intelligence p. 71-101 (2012)

Available from: http://dx.doi.org/10.1007/978-1-4614-3773-4_3

The original publication is available at www.springerlink.com

Accessed from: <http://hdl.handle.net/1959.13/1306073>

Caslav Bozic, Stephan Chalup, and Detlef Seese. **Application of intelligent systems for news analytics**. In Michael Doumpos, Constantin Zopounidis, and Panos M. Pardalos, editors, Financial Decision Making Using Computational Intelligence, volume 70 of Springer Optimization and Its Applications, pages 71-101. Springer US, 2012.

DOI: 10.1007/978-1-4614-3773-4_3

Print ISBN 978-1-4614-3772-7

Online ISBN 978-1-4614-3773-4

```
@InCollection{BozicEtAl2012,
  author = {Caslav Bozic and Stephan Chalup and Detlef Seese},
  editor = {Michael Doumpos and Constantin Zopounidis and Panos~M.
Pardalos},
  title = {Application of Intelligent Systems for News Analytics},
  booktitle = {Financial Decision Making Using Computational Intelligence},
  series = {Springer Optimization and Its Applications},
  volume = {70},
  publisher = {Springer US},
  address = {New York},
  year = {2012},
  pages = {71—101}
}
```

Chapter 3

Application of Intelligent Systems for News Analytics

Caslav Bozic, Stephan Chalup, and Detlef Seese

Abstract The chapter starts with an overview of existing text mining systems whose main purpose is predicting equity price movements on the financial markets. In general, these systems transform the input text to a so called sentiment score, a numerical value equivalent to the opinion of an analyst on the influence of the news text to the further development of the regarded stock. In the second part it is explored how the sentiment score relates to some of the relevant macroeconomic variables. It is suggested that raw sentiment score can be transformed to reveal sentiment reversals, and such transformed indicator relates better to future returns. As an example the project FINDS is presented, as an integrated system that consists of a module that performs sentiment extraction from the financial news, a benchmark module for comparison between different classification engines, and a visualization module used for the representation of the sentiment data to the end users, thus supporting the traders in analyzing news and making buy and sell decision.

3.1 Introduction

News articles make a very important information source for traders. News stories reach a 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

Caslav Bozic
Institute AIFB, Karlsruhe Institute of Technology, Germany e-mail: bozic@kit.edu

Stephan Chalup
School of Electrical Engineering and Computer Science, The University of Newcastle, Australia
e-mail: stephan.chalup@newcastle.edu.au

Detlef Seese
Institute AIFB, Karlsruhe Institute of Technology, Germany e-mail: detlef.seese@kit.edu

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 into new trading algorithms emerged. This task has to be done on a 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.

There are two paths of using quantified text sentiment information - one is feeding it to the machinery that executes trades automatically, and there it is combined with other indicators and used by an algorithm to execute trades most efficiently. The alternative approach is to present this sentiment information to traders visually in a comprehensible way, so the news articles are summarized and the important ones can be filtered. This second approach offers support to professional traders in making trading decisions, and relieve them from the burden of following all news releases, including those unimportant and uninteresting.

In this chapter we will give an overview of the most important systems with this goal found in the literature, and then describe the system developed within the Karlsruhe Institute of Technology (KIT) and its unique way of using Neural Networks and Support Vector Machines methodology to analyze news articles and support traders in their decisions.

3.2 Present Results

In the past decade different systems and methods appeared in the literature that try to solve the task of automatic news analysis. 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. Although some of them do not explicitly use the term ‘text sentiment’, if we observe text mining methodologies as transformations that assign a numerical value to every textual string, we can refer to that numerical value as a sentiment score. All these publications have at least implicit statements about the predictive power of the specific sentiment score on e.g. returns or volatility.

In the following sections we will describe these systems, always following the similar structure. We start with a brief overview and a list of publications where the system was described. Then a source of news data is described, followed by a description of a quantitative data from markets that was used and a definition of target values or classes. Afterwards, we present approaches to document representation, followed by employed classification methodologies. Finally, methods of performance assessment are described, and a section closes with a short discussion, where appropriate. If not stated otherwise, we describe the most recent version of the system, supposing it to be most advanced and best performing.

3.2.1 System *LOLITA for Finance*

Forerunner of all systems of this type that can be found in the academic literature was a financial text summarization system based on natural language engineering system *LOLITA* (Large-scale, Object-based, Linguistic Interactor, Translator, and Analyser). It was developed at the University of Durham and it is first mentioned in [43]. In the same year the first proposal to use this system for financial text analysis came as a publication [14]. The financial application of the system was described in a series of publications [16, 13, 15, 12]. The *LOLITA* system translates text into a semantic network representation, and the built-in templates for financial activities extract specific events that are expected to have an influence on the market movements. Identified and implemented built-in templates can recognize company takeover, merger, privatisation, new issue of shares, new stake, dividend announcement, overseas listing, and bankruptcy. Further analysis and relating extracted event properties to a quantitative data from stock markets were out of scope of this work. That is why it is noted as a forerunner only.

3.2.2 System *PDR*

The first prototype of the system that uses data mining techniques to predict market movements was presented in the master thesis [35]. The extended version was described shortly afterwards in [66]. The system originates from Hong Kong University of Science and Technology, and it employs methodology of probabilistic dialog rules (PDR) to predict daily change of Hong Kong's Hang Seng Index (HSI). The improved system is presented in the subsequent publications [10, 9], and finally it is in detail described in the PhD thesis [11].

The analyzed text is built by merging all articles published during one day in all relevant data sources. The list of data sources contains 41 daily changing documents from financial web sites: Wall Street Journal (www.wsj.com), Financial Times (www.ft.com), CNN (www.cnnfn.com), International Herald Tribune (www.ihf.com), and Bloomberg (www.bloomberg.com). There is an elaborated procedure for selecting relevant data sources by comparing its performance on the training dataset.

The textual data is aligned with daily values of the Hang Seng Index. The total of 120 training days are split into three classes, 'up', 'steady', and 'down', according to daily change of HSI. Changes between -0.5% and $+0.5\%$ are considered as 'steady', while changes of more than 0.5% in positive and negative direction are considered as 'up' and 'down', respectively. The threshold value used in the first version of this system was 0.3% .

The underlying vocabulary is handcrafted, but unfortunately the complete list can be found only in [35] and contains 125 phrases, e.g. 'hang seng index surged', 'shares rose', 'political worries', or 'dollar edged lower'. According to [11] later vocabulary contains 392 words and phrases, selected from the list of 790 suggested

by experts. The first version of the system matched only the complete vocabulary phrases in the analysed text, what is in subsequent version changed to word-level matching, but no version uses word stemming. The document vector is built using term frequency with either inverse document frequency (IDF), or category frequency (CF).

The system uses PDR to build rules that assign scores to text documents. The set of rules is built for each target class, so each analysed document is assigned three distinct scores for ‘up’, ‘steady’, and ‘down’. The scores are compared to certain thresholds to obtain a categorical result (zero or one) for each of the classes. If the result is inconsistent, and the text is assigned to none or more than one class, the conflict resolution is conducted. Three different conflict resolution methods are proposed: based on the simple maximum likelihood, on the maximum deviation from the corresponding threshold, and on the nearest neighbour.

As the final step, the system predicts closing value of HSI HSI_t using classification result CP and previous day’s closing value HSI_{t-1} . Three delta values δ_{up} , δ_{steady} , and δ_{down} are calculated from training data, for each class separately. The delta is defined as an average difference of HSI values in two consecutive days. N_c represents number of days in training set falling into particular class c .

$$\delta_c = \frac{1}{N_c} \sum_{t \in c} (HSI_t - HSI_{t-1})$$

The predicted closing value HSI'_t is then calculated by adding the appropriate delta for the predicted class CP to the previous HSI closing value.

$$HSI'_t = HSI_{t-1} + \delta_{CP}$$

Performance evaluation is consistently done using overall accuracy as the main performance measure. The best reported result in [11, p. 161] is 51.9% accuracy for predicting direction of HSI change and it is achieved with majority voting among multiple data sources and calculated for 79 trading days. Besides HSI, system described in [66] predicts also Dow Jones Industrial Average, Nikkei 225, Financial Times 100 Index, and Singapore Straits Index. The best reported accuracy there is 46.7% for the Financial Times 100 Index. These evaluations are done using 60 trading days.

In one short passage of [10] the same result is expressed in a different manner. The authors analyse the probability that the outcome of classification would have reached certain accuracy if the prediction was totally random. Even for the worst prediction accuracy in the experiment of 40.2% they could refute the null hypothesis that the same result could come from random guessing with 99.82% confidence.

It is worth mentioning that [66] compares the system based on PDR with two further systems based on k-nearest neighbour learning and neural networks. For the feed-forward neural network with 423 input neurons (one for each of the features), 211 neurons in hidden layer and three output neurons, trained by back propagation, they report the best accuracy of 43.9% for HSI, calculated on 40 trading days. Un-

fortunately, further information about the neural network (e.g. activation function and determination of one resulting class) was not included in the publication.

The same article employs a trading strategy for performance evaluation. The articles published overnight are used as an input to the system, as well as the closing index value from the previous day. This allows to predict the direction of index change before the market opening. According to the trading strategy, stocks representing index or appropriate options are bought if the resulting class is ‘up’, and sold if the resulting class is ‘down’. If the system classifies text to ‘steady’ nothing is done. The reported profit for the Dow Jones Industrial Average is 7.5% for 60 trading days where actual trading according to the strategy takes place on 40 days.

As the authors of [42] well noted, the trading strategy is based on the assumption that at the moment the market opens one can on average sell and buy at yesterday’s closing price. They state

However, this approach is not acceptable, because - taken that HKUST (system developed at Hong Kong University of Science and Technology) recognizes the majority of positive respectively negative developments correctly - it is unlikely that in the case of a positive (negative) prediction one can buy (sell) at yesterday’s closing price, but in average certainly at a higher (lower) price.

Maybe even more important is the fact that the authors of this trading strategy do not use real changes in index value for the evaluation, but instead they use the fixed thresholds defined for each of the classes, in this case +0.5% and −0.5%. This means that the conclusion about profitability of the trading strategy is to be taken with reservations.

Table 3.1 Properties of the Prototype PDR

Prototype PDR		
Features definition	handcrafted	
Inputs	text	41 data sources a day
Classifier	rule based - PDR (k-nearest neighbour) (feed-forward NN)	
Output	categorical	
Prediction	index value change	
Frequency	daily	
Performance	overall accuracy	51.9%/3 classes
	trading strategy	7.5%/40 trades
Evaluation data points	40	

3.2.3 *System DC-1 for Finance*

Originally created for a fraud detection, in [22] the authors present an application of the DC-1 system for the task of news monitoring. The paper in which it has been presented as a main contribution has a high-level definition of the general monitoring task which is a part of data mining. The authors propose a framework for evaluating performance of the monitoring systems based on a modified Receiver Operating Characteristic (ROC) curve (for more on ROC see [47, 21]). They use news monitoring task as an illustration of their framework's application.

The evaluation dataset contains news messages published the three months period. The exact period, the total number of messages, and the source of the messages were not disclosed. There are about 6,000 different companies detected in the dataset. The news stream was aligned with (most probably daily) stock prices, and the event of interest was a surge or a drop in the stock price greater than 10%. According to the proposed framework, the scoring function has to be defined. The scoring function represents the utility of detecting the event in a certain point in time, and it is set to 1 if the event of a price change was recognized during the previous day, or on the day of the price change before 10:30 am. Otherwise, the scoring function is set to 0.

Each story was lexically analyzed and reduced to its constituent words. The words are stemmed and those from a stop word list are removed from the further analysis. The ordered pairs of two adjacent words (bigrams) were also used in the final vector representation of the news story. The stories preceding substantial change in price (either positive or negative) are labeled as positive examples. This means that all subsequent performance analyses do not distinct between positive and negative change in stock prices, so any direct economic interpretation of the results is difficult.

For performance evaluation the authors use a modified version of ROC named Activity Monitoring Operating Characteristic (AMOC). The variables represented on the axis of this graphical representation are modified to be in accordance with the sequential nature of the monitored data streams. So the x axis represents false alarm rate, normalized to some time interval, and y axis shows average score, according previously defined scoring function. The results presented in the paper are calculated using 10-fold cross-validation. The AMOC curve of the system is presented in the publication, and some of the specific points read from the graph are e.g. (0.1,0.76), (0.2,0.89), and (0.5,0.97). The points of the AMOC curve are determined by varying the threshold of the event detection trigger.

The authors introduce two orthogonal distinctions for classifying approaches in stream mining: the first distinction separates 'profiling' on the one end, where only one class of the events is modelled, and all the events that substantially differ from this profile are considered events of interest and 'discriminating' on the other end, where both classes are modelled with specific attention to differences between classes. The other distinction is made between a 'uniform' and an 'individual' approach. In the 'uniform' approach only one model for all streams is built, while 'individual' approach models each data stream individually. The authors show that

the individual approach does not bring much improvement in the news monitoring case study, although introducing more complexity.

Table 3.2 Properties of the Prototype DC-1 for Finance

Prototype DC-1 for Finance		
Features definition	automatic	
Inputs	text	three months & 6,000 companies
Output	categorical	
Prediction	stock price jump	
Frequency	(daily)	
Performance	ROC-AMOC	
Evaluation data points	undisclosed	

3.2.4 System *Æ*nalyst

Started at the University of Massachusetts Amherst, the system under the name of *Æ*nalyst aims at identifying news stories that strongly correlate with significant shifts in the stock price. The project proposal is published in [32], while the actually implemented system was described in [33] and [34]. The system uses news messages collected from Biz Yahoo! and modified time series of stock prices transformed to trends by piecewise linear regression, and tries to mine these two types of data concurrently.

The source of the data are news stories published via Biz Yahoo! in the period from October 15th 1999 to February 10th 2000. The authors report 38,469 news stories mentioning 127 companies, and the stories are already tagged with the company names they mention. The selection of the companies is done according to the specific criteria: 15 stocks with the largest increase on any day, 15 with the largest decrease on any day, and the certain number of stocks from the three lists published periodically that mention companies that are most actively traded, gainers, and losers. This kind of ex post selection of the companies comprising the dataset makes the trading strategy evaluation results biased in the way that is not attainable in the real market conditions (see [42]).

The quantitative part of the dataset contains stock prices for these 127 companies during the opening hours of the exchanges, with the 10 minutes resolution. The period is equivalent to the period of the news dataset. ‘Piecewise linear regression’ is used to represent the time series as an array of trend lines. These trends are clustered according to the slope and confidence (squared error) of each linear segment, and after the inspection of the distribution, it is decided that a simple binning procedure can be used. The trends are divided into five classes: ‘surge’ (‘plunge’), if the slope

is greater (less) than 75% of the maximum (minimum) slope, ‘slight+’ (‘slight-’) if the slope is between 50% and 75% of the maximum (minimum) observed slope, and ‘no recommendation’ otherwise.

The authors propose named entities, linked objects, and subsumption hierarchies as approaches for text representation (see [32] for more details). Though, the evaluation in subsequent publications [33, 34] is done for bag of words approach only. Bag of words is an approach to document representation where unordered collections of words are used, assuming conditional independence and disregarding the order of words. The system builds a language model for all messages that precede the beginning of the particular trend for a certain period. If t is the moment in which the particular trend starts, then all the messages published between $t - h$ and t are considered to have influence on the appearance of the observed trend. These messages are used to build the language model for the class of the observed trend. The variable h represents the chosen time horizon and it is varied from 5 to 10 hours during the evaluation. Authors use additional approach of aligning only the stories published contemporaneous with the observed trend, with the note that this kind of model is rather explanatory than predictive.

After all of the news stories are associated with trends, and trends are divided into classes, classification is done by building language models with an assumption of words independence, effectively arriving at a naïve Bayesian classification. Since the news analytics and news classification is an on-line task, where used language can change, we need to assure that these new terms have non-zero probability, otherwise it would drive the model’s probability to zero value. This is done by a linear back-off algorithm, where the conditional probability of word ω under the model M_t is given by formula

$$P(\omega|M_t) = \lambda_t P_{ml}(\omega|M_t) + (1 - \lambda_t)P(\omega|GE)$$

where P_{ml} represents the original probability under the certain language model that possibly can be equal to zero, and GE denotes background of General English language model. The parameter λ_t is calculated according to formula

$$\lambda_t = \frac{N_t}{N_t + U_t}$$

with N_t denoting total number of tokens and U_t denoting the number of unique tokens in the model M_t .

The Evaluation of the prototype is done using two approaches: a Detection Error Tradeoff (DET) curve, similar to ROC, and a trading strategy. DET is a graphical representation of the classification performance that plots miss rate against false alarm probability. 10-fold cross-validation was used to calculate the results. The plot contains points like (0.5,90) and (15,60) being percentages of false alarms and miss rates, respectively. This means that the system can achieve 10% recall with false alarm rate of 0.5% in the task of predicting exact class of the trend up to 10 hours ahead.

The previous evaluation works with pairs (t,D), where t represents a trend and D a document, so one document can be associated with more than one trend (in fact it is associated with all trends that follow the particular document in next 10 hours). For a practical evaluation, one need to introduce conflict resolution, and determine only one prevailing resulting class for the each document. This is done in the evaluation that uses trading strategy. Different language models are built for each stock using the data from the first tree months of the dataset, while the rest of the dataset was used for evaluation. According to the most probable class, the trading strategy should buy or short sell the stock worth of 10,000 USD. The budget is considered unlimited and the transaction costs are considered to be zero. The reported total gain was 21,000 USD within the whole dataset, with e.g. for the case of Yahoo stocks 0.5% gain per transaction. With changing the time window for associating documents and trends in the range form 1 to 10 hours, the average profits varied from 290 USD to 2,360 USD.

Table 3.3 Properties of the Prototype ÆAnalyst

Prototype ÆAnalyst		
Features definition	automatic	
Inputs	text	Biz Yahoo!
Classifier	naïve Bayesian	
Output	probabilities/categorical	
Prediction	trend class	
Frequency	10 minutes	
Performance	DET	
	trading strategy	0.5%/trade
Evaluation data points	12,174 trades	

3.2.5 System from University of California, San Diego (UCSD)

News analytics prototype developed at the University of California in San Diego and financed by United States Airforce was described in a project report [26] and somewhat more detailed later in [25]. In the first version it uses the same dataset as prototype ÆAnalyst and an original approach to aligning text and price values. The naïve Bayesian classifier is used for text classification.

The dataset contains 25,087 articles about 30 companies of the Dow Jones Industrial Average (DJIA) index in the period from July 26th 2001 to March 16th 2002. After excluding the articles published outside of business hours, and excluding the articles that correspond to the periods where price data is incomplete or missing,

the authors report 12,437 news articles. The influence of the published article to the market is modeled by the ‘window of influence’ this creating the model similar to event study. Window of influence is the period in which the published article should have the influence on the stock price of the company that is mentioned. Different lengths of the window of influence up to 30 minutes are evaluated, and this is done for the two cases: when the window of influence ends at the moment of news article publication, and when the window of influence starts at the moment of news article publication.

The price data is collected on the one minute level. The classes are determined according to the β and index adjusted change m in the stock price within the window of influence. The price data is transformed according the following formula:

$$m(t_1, t_2) = \frac{\delta p_s(t_1, t_2)}{\beta} - \delta p_i(t_1, t_2)$$

where t_1 and t_2 represent endpoints of the window of influence, $\delta p_s(t_1, t_2)$ is the change in the observed stock price and $\delta p_i(t_1, t_2)$ is the change of the index value during the window of influence. The parameter β determines relation between the price of the particular stock and the market as a whole. The value of DJIA index is chosen as a proxy for market performance in this case.

According to this value the target classes are labeled as ‘up’, ‘normal’, and ‘down’. If the value of m is less than negative threshold $\rho_{negative}$, the target class is ‘down’, if the value of m is greater than positive threshold $\rho_{positive}$ the target class is ‘up’, and otherwise the target class is ‘normal’. The values for $\rho_{negative}$ and $\rho_{positive}$ are determined in such a way that the frequencies of the classes ‘up’, ‘normal’, and ‘down’ are approximately 25%, 50%, and 25%, respectively. The classification is performed using a naïve Bayesian classifier.

Depending on the prior distribution of classes the trivial classifier that always predict the class with highest prior can have better overall accuracy, although not being economically useful, as noted by the authors. That is why they suggest two performance measures based on the economic utility of the classification. One is based on the β and index adjusted price change m , which authors call ‘normalized economic value estimate’ (NEVE). They also mention the major drawbacks of this measure by saying

...the meaning of NEVE is not easy to understand, nor even its unit is clear.

The second performance measure the authors suggest is the average profit per trade achieved in the specific trading strategy. According to the strategy at the beginning of the window of influence one buys (sells) 1,000 USD worth shares of the mentioned company and sells (buys) $\beta \times 1,000$ USD worth of the index if the predicted class was ‘up’ (‘down’). At the end of the window of influence this position is cleared.

The best performance according to both criteria is achieved using the window of influence with the length of 20 minutes before the publication of the news article. The reported profit per trade is 1.0063 USD and NEVE value of about 0.17. The question of economic applicability of the trading strategy in real conditions stays

open, since it seems that it incorporates some actions that have to take place before the classification of the news article is performed.

Table 3.4 Properties of the Prototype UCSD

Prototype UCSD		
Features definition	automatic	
Inputs	text	
Classifier	naïve Bayesian	
Output	categorical	
Prediction	normalized price change	
Frequency	1 minute	
Performance	NEVE	0.17
	trading strategy	1.0063 USD/trade
Evaluation data points	13,372 articles	

3.2.6 System from the Chinese University of Hong Kong (CUHK)

The prototype developed at the Chinese University of Hong Kong is the first system which analyses news stories published by Reuters, thus achieving to collect about 600,000 news items for the training and evaluation. It is mostly composed of already existing natural language processing (NLP) and machine learning (ML) components, combined in a new manner to allow for parallel mining of text and multiple time series. The system is described in two publications [23, 24].

These 600,000 news articles were published between October 1st 2002 and April 30th 2003, and first six months were used for training, while the last month was used for the out of sample evaluation. The news articles are tagged with company names mentioned, so the system uses all messages in that period that mention 33 components of Hang Seng Index.

The time series of daily prices is transformed using piecewise linear regression, similar to the one in [34]. It is just not clear whether both opening and closing prices represent points in time series, or just one type of daily price, and in the latter case, which one. All trends generated this way are characterized by slope m and goodness of fit R^2 , and clustered into three classes: ‘rise’, ‘steady’, and ‘drop’. The cluster with greatest average slope is selected for the class ‘rise’, the cluster with the lowest average slope is selected for class ‘drop’, while the third cluster becomes class ‘steady’.

The documents are represented by a bag of words approach where all the words are stemmed, and with punctuation, stop words, web links, and numbers excluded. The features produced that way are weighted according to the $tf*idf$ schema.

The authors chose to align the documents with contemporaneous trend only, not allowing for delayed influence of the news to stock prices. They back up this decision by effective market hypothesis, although in later evaluation of the system's performance, they use trading strategy and show its profitability, thus violating the previous principle. The improvement and the novel approach is to once more cluster the documents falling into the categories 'rise' and 'drop', thus creating two subclusters each, and then removing one subcluster from the 'rise' category which is more similar to the documents of the 'drop' category, and vice versa - removing one of two subclusters from the 'drop' category that is more similar to the documents of the 'rise' category. This provides two more distant classes of documents for 'rise' and 'drop' categories, creating less confusion during the learning phase.

The model learning is done using SVMlight Support Vector Machine software [30]. Two SVMs are trained, for 'rise' and 'drop' class. If the news article is classified positive by both of the classifiers, it is labeled as ambiguous, effectively falling into the 'steady' class.

The evaluation is performed by applying a trading strategy to the prices and news data from April 2003. According to the strategy, the shares are bought (sold short) in the moment of publication of the news article that is classified as 'rise' ('drop'), and the position is cleared after 24 hours. If the news article is published during the holidays, the shares are bought (sold short) immediately after the stock market opens on the first business day. Since the authors have only daily data at disposal, it is not clear which daily price they use for the market simulation. The best performing version of the system achieves a rate of return of 0.1985, but the total number of trades was not disclosed. The usual assumption of zero transaction cost applies also.

Table 3.5 Properties of the Prototype CUHK

Prototype CUHK		
Features definition	automatic	
Inputs	text	
Classifier	SVM	SVMlight
Output	categorical	
Prediction	trend	
Frequency	daily	
Performance	trading strategy 0.1985 rate of return	
Evaluation data points	1 month daily data for 33 companies	

3.2.7 System *AZFinText*

One of the recent systems that is still in development bears the name of *AZFinText*. It is built upon one version of Arizona Text Extractor (*AzTeK*) which is described in [64]. The *AZFinText* system aims at predicting price of particular stock after a 20 minutes period by analyzing news published via Yahoo Finance. The system is presented consistently and with slight constant improvements in a long row of publications [50, 51, 58, 56, 55, 52, 53, 57, 54].

As the only data source the system uses articles from Yahoo Finance, tagged with the company they relate to. Texts published outside of stock markets opening hours, as well as those published less than one hour after markets opening and less than 20 minutes before markets closing, are filtered out. Excluded are also messages that are less than 20 minutes apart. The final dataset used for analysis contains 2,802 articles from 5 trading weeks that relate to 484 of the S&P 500 companies. The goal is to predict the exact equity price exactly 20 minutes after the news text has been published.

The input text is preprocessed using four different approaches: bag of words, noun phrases, proper nouns, and named entities. Only phrases with three or more instances per document are represented with one in resulting binary vector, while the other phrases are represented as zero. The binary document vector created in this way, together with equity price in the moment of article publication, is used as an input to machine learning engine.

Machine learning part of the this prototype is built on SVM trained using sequential minimal optimization as described in [46] and using linear kernel. As an output from the training step, SVM estimates parameters of a regression by assigning weights to each of the input terms. The dependent variable of this regression represents the predicted equity price.

In [58] the authors use additional inputs to the machine learning engine. It represents sentiment assessment of an analyzed article, produced by *OpinionFinder*. *OpinionFinder* analyses text and classifies it as rather 'subjective' or rather 'objective', and its tone as 'positive' or 'negative'. In both classifications the classifier can also abstain from classification. The article reports that just adding subjectivity and tone information leads to deteriorated result, but on the other hand, results improve when the prediction is done only on subset of articles classified by *OpinionFinder* as 'subjective' or 'negative'.

For performance evaluation there are three approaches consistently used through publications. All of them employ 10-fold cross-validation. The first measure of accuracy is equivalent to the mean squared error between real and predicted equity price. The lowest reported value is 0.03407 on 2,620 data points when named entities preprocessing approach is used. Unfortunately the nature of this measure and its dependence on absolute value and range of target variables makes it unfit for further comparisons.

The second approach considers observing the direction of change only. If a predicted price and a realized price are both higher or both lower than the price at the moment of prediction, the prediction is considered correct. This is equivalent to the

overall accuracy for the case of two-class classification. The best reported value in this category is 58.2% which is achieved using a proper nouns preprocessing approach on the whole test set of 2,809 data points, with higher accuracy of 59% if calculated for 61 subjective text only and even higher for the case of training the SVMs separately for different industry sectors, reaching 71.18%.

The third approach is using a trading strategy that follows ideas given in [40]. It invests by buying stocks which have predicted an increase of more than 1%, and short-selling a stocks which have predicted a decrease of more than 1%. The position is in both cases cleared after 20 minutes. The best reported profit 8.5% is reported when using SVMs trained separately on different industry sectors in [55]. It is, though, not clear how many trades this scenario contains. The number of trades is only reported in [50], where the best profit of 3.6% is achieved in 108 trades with named entities preprocessing model used.

Table 3.6 Properties of the Prototype AZFinText

Prototype AZFinText			
Features definition	automatic		
Inputs	text	Yahoo Finance	
Classifier	SVM		
Output	categorical		
	real		
Prediction	stock price		
Frequency	1 minute		
Performance	mean squared error	0.03407	
	overall accuracy	71.18%	
	trading strategy	8.5%	
Evaluation data points 10-fold cross-validation on 2,802 articles			

3.2.8 Other Systems

Further prototypes include the system called News Categorization and Trading System (NewsCATS) described in [42, 41]. NewsCATS is a high frequency forecast system that classifies press releases of publicly traded companies in the U.S. using a dictionary that combines automatically selected features and a hand-crafted thesaurus. For classification the authors use SVMs with polynomial kernels.

One of a few systems that employ neural networks is described in [36, 37]. The system proposed in the first publication uses only the volume of posted internet stock news to train a 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 [37] employs natural language processing techniques and a hand-crafted dictionary to predict stock returns. They use a feed-forward neural network with five neurons in the input layer, 27 in the hidden layer, and one output neuron. Since only 500 news items were used for the analysis, no statistical significance of the results could be found.

Another group of publications contains works that do not primary attempt to prove economical relevance of published text by evaluating specifically tailored trading strategies, but rather to find statistically relevant relations between financial indicators and sentiment extracted from the text.

The authors of [1] use naïve Bayes and SVM classifiers to classify messages posted to Yahoo!Finance and Raging Bull and determine their sentiment. They do not find statistically significant correlation with stock prices, but they find sentiment and volume of messages significantly correlated to trade volumes and volatility. In their methodological paper [18] authors offer a variety of classifiers, as well as composed sentiment measure as a result of voting among classifiers. In the illustrative example they analyze Yahoo stock boards and stock prices of 8 technology companies, but they do not find clear evidence that the sentiment index can be predictive for stock prices.

There are two pivotal articles published in the Journal of Finance. Tetlock in [61] observes Wall Street Journal's column "Abreast of the Market", uses content analysis software General Inquirer together with principal component analysis approach and finds that high pessimism in published media predicts downward pressure on market prices. The authors of [62] succeeded to find that rate of negative words in news stories about certain company predicts low earnings of the company.

While most of the previously described publications focus on predicting price trends of single stock or index, there are publications that aim at determining influence of news releases to volatility. System [63] improves the risk-return profile by exiting the market in case of news that are predicting high volatility, while [49] attempt to classify press releases of German public companies according their influence on volatility of stock prices. There are also systems that aim at predicting foreign exchange rates, e.g. [45] and [65], but we will stay focused on the sentiment extraction systems for equity prices prediction.

From the previous sections and Tables 3.1 - 3.6 we can notice that existing systems implement various classification methodologies – at the beginning mostly naïve Bayes and decision rules, later more SVMs, and recently just a few that use neural networks. The feature definition procedure and document representation are quite different between systems, what makes comparison of different classification approaches a complicated task.

It is of special interest to compare the performance evaluation of all these systems. Some of performance measures are purely machine learning and statistically oriented, making them less useful for financial setting. The others are based on historical data and trading strategies, but in most of these cases lack enough evaluation data points, or use assumptions that are unattainable in real life. All presented systems have between 40 and about 13,000 evaluation data points, what might be not

enough to guarantee stability of the results. Additionally, both systems having more than 10,000 evaluation points include some assumptions that are observable only ex post.

We suggest to bridge this gap by creating a benchmark based on sufficient data to ensure statistical relevance of the results (over 100,000 articles and 600,000 daily returns). We also offer the spectrum of different classification approaches, while keeping the text preprocessing and document representation constant, enabling the effective comparison of classification engines.

Besides the prototypes described in the academic literature, there are systems developed in the industry. They employ methodology that is proprietary and rarely described in the academic literature. Nevertheless, they represent de facto industrial standard, being used by a great number of professional traders. Such services are provided under the name of News Analytics in the product portfolios of companies like RavenPack (see Section 3.3.2) and Thomson Reuters (see Section 3.3.1).

3.3 Text Sentiment Properties

The publications presented in the previous section showed a relation between extracted sentiment from financial news and equity prices, trade volumes, and volatilities. We will take here the macroeconomic view and explore the influence of such sentiment score to GDP and other macroeconomic variables. We also propose a way of transforming raw sentiment score to expose sentiment reversals and show that the data transformed in this way relate even better to future returns.

3.3.1 *Macroeconomic Data*

The data used in this research originates from two main sources: the source for quantified news data and the basis for sentiment score comes from Reuters NewsScope Sentiment Engine for historical data, while the public World Economic Outlook (WEO) database [29] of the International Monetary Fund provided the macroeconomic data.

Data constituting the output of the Reuters NewsScope Sentiment Engine (now part of the Thomson Reuters News Analytics product) represents the authors sentiment measure for every English-language news item published via NewsScope in the period from 2003 to 2008 inclusive. The measure classifies a news item into one of three categories: positive, negative, or neutral. The probability of the news item falling into each of the categories is also given. Each record represents a unique mention of the specific company, with a possibility of one news item relating to more than one company. In our dataset there are 6,127,190 records. Each news item can have multiple tags called topic codes, and topic codes are grouped into categories. One of the categories represents countries, and it can be used for determining what

country is mentioned in the particular news item. Using this tagging feature, average sentiment per country can be calculated. The data is aggregated using the system described in [3] to the level of country and year. The data provided by Reuters are outputs of the classifiers in the form of three probabilities: probability that the analysed news item is positive in sentiment P_{pi} , that it is neutral P_{oi} , or that it is negative in sentiment P_{ni} . We define sentiment score S_i by subtracting probabilities for positive and negative class, and by aggregation we get a yearly sentiment score for a country $S_{y,c}$.

$$S_i = P_{pi} - P_{ni}$$

This way we get panel data with six consecutive years and 233 countries and territories. We combine this data with the data for the same period from the WEO database. After excluding countries that are not a part of the WEO database, we get the data about 181 countries. The selected subset of variables from WEO dataset is given in Table 3.7. The estimation of panel regression coefficients on the data is performed with the help of OxMetrics and DPD software [20].

Table 3.7 Selected variables from WEO dataset

Variable Description	Units	Scale
Gross domestic product, constant prices (change)	Percent change	
Gross domestic product, current prices	U.S. dollars	Billions
Gross domestic product per capita, current prices	U.S. dollars	Units
Gross domestic product based on purchasing-power-parity (PPP) valuation of country GDP	Current international dollar	Billions
Gross domestic product based on purchasing-power-parity (PPP) per capita GDP	Current international dollar	Units
Gross domestic product based on purchasing-power-parity (PPP) share of world total	Percent	
Investment	Percent of GDP	
Gross national savings	Percent of GDP	
Inflation, average consumer prices	Index	
Inflation, average consumer prices (change)	Percent change	
Inflation, end of period consumer prices	Index	
Inflation, end of period consumer prices (change)	Percent change	
Import volume of goods and services (change)	Percent change	
Export volume of goods and services (change)	Percent change	
Unemployment rate	Percent of total labor force	
Population	Persons	Millions

Table 3.8 shows estimated values of coefficients for the OLS panel regression when only contemporaneous sentiment is used as independent variable. We find positive and significant relation of the average sentiment of all news items published in one year and mentioning a country, with that country's GDP in that year, and with that country's GDP change compared to the previous year. Thus, the country with more positive news published in a year will have greater GDP and more positive GDP change by the end of the year.

Table 3.8 Estimated coefficients, contemporaneous sentiment score as an independent variable

Variable Description	Coefficient
Gross domestic product, constant prices (change)	1.5138 **
Gross domestic product, current prices	44.3611 **
Gross national savings	6.2975
Import volume of goods and services (change)	4.6756
Export volume of goods and services (change)	1.8859
Population	1.1836 ***

This prediction goes beyond the time span of one year, as the data from Table 3.9 suggests. It shows that the sentiment score in one year can be significantly and positively related to GDP, GDP change, gross national savings, and import volume for the following years, up to three years ahead. The negative estimations for the coefficients in Table 3.9, like relation to export volume, or import volume three years ahead, are all not significant, so we can not make any statements on that relations.

Table 3.9 Estimated coefficients, lagged sentiment score as an independent variable

Variable Description	S	$S(t-1)$	$S(t-2)$	$S(t-3)$
Gross domestic product, constant prices (change)	2.8990**	1.3739	1.6556*	0.9656
Gross domestic product, current prices	32.6456	67.7682*	52.5004	93.2768***
Gross national savings	14.6525*	14.6006**	8.5230*	6.2042**
Import volume of goods and services (change)	13.8133**	12.5976**	0.4568	-2.9750
Export volume of goods and services (change)	1.4063	-4.2891	-2.9645	-0.9618
Population	0.9589***	0.4708	1.0382**	1.2311***

3.3.2 *Sentiment Reversals*

Since the scientific exploration of text mining and sentiment extraction in finance lasts for over a decade, as noted in [7]

...simple “buy on the good news” and “sell on the bad news” strategies won’t likely generate significant alpha as news analytics become more widely adopted.

That is one of the reasons for the great number of publications in recent years which try to find novel approaches and exploit sentiment scores provided by existing systems.

One of the recent publications, with an approach similar to ours, is [31]. The author uses sentiment reversals as buy signals, and proves the performance of his approach by simulating a set of portfolios over the time frame 2000-2008. The source of sentiment data is Dow Jones News Analytics historical database, where the sentiment scores have been assigned to news articles using RavenPack proprietary software. Each story is assigned a positive or a negative sentiment indicator as a value of +1 and -1, respectively. Each story is also related to the company mentioned in the text. This approach builds one time series of sentiment indicators for each company. The author uses an average value of the sentiment time series over a certain time window to define sentiment reversal. Sentiment reversal is, in fact, defined as an end of the period in which the running window average sentiment has a constantly positive (negative) value. Graphically, this corresponds to all points where graphical representation of the running window average sentiment intersects the x axis. There are some additional constraints introduced, like minimal duration of reversal period of 30 days, and minimally 30 published stories in this period.

The author’s findings suggest that the portfolios built using sentiment reversal based signals for buying and short selling the companies’ stocks can outperform the market both in bull and bear market conditions.

We explored a similar approach, although using different dataset and basing our performance assessment on more general and theoretical approach described in [3] instead of portfolio simulation.

The question we want to answer here is whether sentiment data transformed by extracting sentiment reversals can be more useful for predicting future returns for a company than the raw sentiment score data. We work with sentiment data provided by Thomson Reuters, and evaluate it according the framework described in [3].

Historical database of Thomson Reuters News Scope Sentiment Engine is a source of sentiment scores for all English language news published via Reuters News Scope system since year 2003. For each company mentioned in the news, this database contains the probabilities that the author write about the particular company in rather positive or rather negative context. The sentiment score $S(n, c)$ for news article n and company c is built as a difference between the probability that the mention is positive (Pr_{pos}) and the probability that the mention is negative (Pr_{neg}). Since we want to work with daily data, the sentiment score is averaged within each trading day, thus producing daily sentiment score for a company $S(t, c)$. With T we denote a set of all messages published during the trading day t .

$$S(n, c) = Pr_{pos}(N, C) - Pr_{neg}(N, C)$$

$$S(t, c) = \sum_{n \in T} S(n, c)$$

Besides sentiment scores this dataset offers additional metadata. Most important for us are the publication time stamp 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 those news items related to companies that are constituents of the Russell 3000 index. The Russell 3000 Index consists of the largest 3,000 U.S. companies representing approximately 98% of the investable U.S. equity market.

The sentiment reversal measure is defined in the following way - for each company a running average sentiment score $S_{avg}(t, c)$ is calculated for each day t . In the moment of news article publication, instead of raw sentiment score as defined in the previous paragraph, we assign a new value that measures sentiment reversal, or sentiment deviation from the average value S_r . The number of total trading days before the current day t is denoted by N_t

$$S_{avg}(t, c) = \frac{\sum_{t_l < t} S(t_l, c)}{N_t}$$

$$S_r(t, c) = S(t, c) - S_{avg}(t, c)$$

As a source of trading data we used Thomson Reuters Tick History database. We extract opening prices for all trading days in 2003 for each company from the Russell 3000 index. The opening prices are adjusted for dividends and then transformed into log-returns. In this way we get open-to-open (R_{OO}) returns for each trading day in 2003 and each Russell 3000 company. The respective equation is given below, where P_O represents opening stock price and t represents the current trading day.

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

We next align daily sentiment data (both raw data and reversal measure) with daily returns, and build the linear regression to explore relation between these variables. If the observed 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 3.1), 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 daily sentiment score $S(t)$, daily sentiment score from day before $S(t-1)$, two days before $S(t-2)$, and three, four, and five days before $S(t-3)$, $S(t-4)$, and $S(t-5)$. 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.

$$R_{OO}(t, c) = \alpha_0 S(t, c) + \alpha_1 S(t-1, c) + \alpha_2 S(t-2, c) + \alpha_3 S(t-3, c) \\ + \alpha_4 S(t-4, c) + \alpha_5 S(t-5, c) + \sum_{i=1}^9 \beta_i dd_i(c) + \gamma \quad (3.1)$$

We repeat the estimation of the regression parameters using sentiment reversal measure $S_r(t, c)$ instead of raw sentiment score $S(t, c)$ following the Equation 3.2. The results are presented in Table 3.10 which offers a possibility for a comparison of these two approaches.

$$R_{OO}(t, c) = \alpha'_0 S_r(t, c) + \alpha'_1 S_r(t-1, c) + \alpha'_2 S_r(t-2, c) + \alpha'_3 S_r(t-3, c) \\ + \alpha'_4 S_r(t-4, c) + \alpha'_5 S_r(t-5, c) + \sum_{i=1}^9 \beta'_i dd_i(c) + \gamma' \quad (3.2)$$

Table 3.10 Estimated OLS coefficients for raw and transformed sentiment score, R_{OO} as an independent variable

Time lag	Raw sentiment score	Sentiment reversal
t	104.159***	100.850***
$t-1$	22.052**	34.596***
$t-2$	-9.126**	-5.034
$t-3$	-8.236**	0.173
$t-4$	-3.569	-2.887
$t-5$	-2.863	0.910

From Table 3.10 it is visible that both sentiment measures have positive and significant relation to contemporaneous return as well as to the return one day ahead. With increasing the lag between sentiment measure and return, this relation becomes less significant for both sentiment measures. Since contemporaneous sentiment does not introduce any incremental forecasting ability, we are more interested in lagged sentiments. Relation between today's sentiment reversal and tomorrow's return is stronger and noticeably more significant than the relation between today's raw sentiment score and tomorrow's return.

The conclusion from the results presented in this section, as well from the previous work is that sentiment reversal measure can be a very useful dimension in news analytics, and that any system aiming in representing the results of the sentiment extraction should not omit representing this parameter.

3.4 Project FINDS

Financial News and Data Service (short FINDS) is a project going on within Information Management and Market Engineering (IME) Graduate School at the Karlsruhe Institute of Technology (KIT) with the goal to conduct innovative research on the analysis of quantitative and qualitative information from financial markets [2]. The topics covered are broad, and include detecting novelty of the published financial articles employing semantic technologies [38, 39], as well as exploring financial markets' reaction to the introduction of machine readable reports in XBRL format [67]. Contribution to the evidence that published information influence financial markets, taking the perspective of empirical microeconomics, has been brought by [60, 48]. The impact of the public announcements on prices of CO₂ emission permits is investigated in [28]. The research presented in the previous section [4, 6, 5] is a part of this corpus, too.

The FINDS system itself consists of three main components - the first component is a natural language processing and text classification engine which is used to extract sentiment from text documents and news articles. Multiple engines exist, and they are based on Bayesian classifier, support vector machines, and neural networks. The extracted sentiment score is delivered to the second component which is used for performance assessment. Employing the methodology of empirical finance, and aligning sentiment data with quantitative data on an extensive dataset, this benchmark can determine relative quality of different classifying engines, and help in choosing the appropriate one for the specific task, as described in [3]. The produced sentiment scores are numerical values that summarize the tone of the great amount of news articles for a number of different companies. To render this numerical summarization useful, we developed FINDS visualization component as a third main component of the FINDS system.

3.4.1 Sentiment Extraction

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 time stamp 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. As mentioned earlier, the Russell 3000 Index consists of the largest 3,000 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.

$$\begin{aligned} R_{OC} &= \ln \frac{P_O(t)}{P_C(t-1)} \\ R_{OO} &= \ln \frac{P_O(t)}{P_O(t-1)} \\ R_{CO} &= \ln \frac{P_C(t)}{P_O(t)} \\ R_{CC} &= \ln \frac{P_C(t)}{P_C(t-1)} \end{aligned}$$

For training we singled out news about four companies: Apple Computer Inc., International Business Machines Corp., Microsoft Corp., and Oracle Corp. We kept only the paragraph where the subject company is mentioned and four surrounding paragraphs, since this amount of surrounding text was proven in previous tests to yield the best performance. 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 11,781 distinct words in the 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 days 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 first classification method that can be trained and then later used for classifying unseen documents is Bayesian classification. It is based on the Bayes theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

If we substitute probabilities of a document falling into a certain class, and look for the class with the highest probability, the equation has the form

$$C = \arg \max_{c \in \text{classes}} \prod_{\omega \in D} \frac{P(\omega|c)P(c)}{P(\omega|B)}$$

With C we denoted the selected class, ω is individual word in a document that we want to classify D , while B represents background distribution of words in a particular language, English in our case. This approach assumes mutual independence of individual words, what is a common assumption in the text classification area. Although this assumption is not strictly correct for most of the languages, there are evidence that this simplification has little influence on practical results [19].

The second classifier is based on support vector machines. As a kernel function it uses radial basis function in the form

$$RBF(x_i, x_j) = \exp -\gamma ||x_i - x_j||^2, \gamma > 0$$

To find the best performing combination of parameters γ and C , which is the penalty parameter of the error term, we perform grid search. Different combination of these two parameters are tested using five-fold cross-validation on the training dataset. The resolution is gradually increased, so we generate the accuracy contour, and can select the best values for the parameters. For the detailed discussion on support vector machines applications in text classification and applications of kernel methods in finance, the reader is advised to consult publications [30] and [8].

The third system uses a feed-forward neural network with an input layer, two hidden layers, and an output layer. The size of the 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 11,781. The two hidden layers consist of 16 and 8 neurons, while output layer has one or two neurons, depending on the version of neural network. We compared two different versions of neural networks. 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.

3.4.2 Benchmark

Each of 107,266 news items published via Thomson Reuters 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.

$$R_{OO}(t, c) = \alpha_0 S(t, c) + \alpha_1 S(t-1, c) + \alpha_2 S(t-2, c) + \alpha_3 S(t-3, c) + \sum_{i=2}^{10} \beta_i d d_i(c) + \gamma \quad (3.3)$$

At this point we need the way to determine the 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 3.3), 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 a contemporaneous text sentiment value $S(t)$, a text sentiment value from day before $S(t-1)$, from two days before $S(t-2)$, and from 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 the company’s size.

Table 3.11 Estimated OLS coefficients with R_{OO} as an independent variable

	Bayes	SVM	NN v1	NN v2	Reuters NSE
sent(-1)	33.22*	44.81*	13.69*	8.96	21.87**
sent(-2)	-10.63***	-10.52	-6.36***	-4.07**	-9.37**
sent(-3)	7.59***	0.840	10.78***	-0.80	-8.79**

The results presented in Table 3.11 reveal that the highest estimated coefficient for the sentiment score is achieved by an SVM, but the most statistically significant result in predicting returns one day ahead has the proprietary Reuters NewsScope Sentiment Engine (NSE). The remaining estimated coefficients are not shown in the table and do not influence predictive power of the systems. All the company’s size dummies and estimated constant are statistically significant to the level of 1%. The star notation of the statistical significance of the results is explained in Table 3.12.

Table 3.12 Statistical significance of the results

	p value
***	< 1%
**	< 5%
*	< 10%
otherwise	$\geq 10\%$

Based on our findings presented in the previous sections, and on the previous works in related areas of text sentiment visualization, we developed a visualization component as a third part of the project FINDS. Visualization of data extracted from great amounts of text is a task being usually solved as a part of the wider project, and often without extensive exploration of previous work in this area. All existing works cluster around two main focuses: topic detection and sentiment extraction from user opinions. For the extensive overview of other prototypes and approaches, the reader is advised to consult surveys [59, 44].

When a news article arrives, it is analysed by one of the FINDS classifiers, or Reuters News Score Sentiment Engine, and the sentiment score for each of the companies mentioned in the article is calculated as a real number between -1 and 1 . The values near -1 indicate news that are rather bad for the affected company, and implicitly signalize negative pressure on company's equity price and returns. As we learned from the previous chapters, sudden change of the sentiment are even better signals of this negative pressure, so when negative news arrives for the company that had in average positive sentiment, that is a strong signal that the downward pressure on prices will be strong in a short future period. This is why our visualization system is designed in a way to make such and similar cases clearly noticeable.

The typical screen of the FINDS visualization component is shown in Figure 3.1. The users have a possibility to set a list of all companies they are interested in, so only these companies will be displayed. Each company is represented by the coloured circle. We can vary four dimensions for each company: x axis position, y axis position, size and color.

The size of the circle representing a company and its sentiment is defined in such a manner that the visibility is kept also in the case of simultaneous representation of a great number of companies. The circle area is linearly dependent on number of messages published about a particular company, and the total area of all displayed representations is kept constant. Using this representation approach, the companies that draw more attention and are mentioned in more news stories are represented with a larger circle.

The color of the company's representation is chosen from the palette that ranges from green, over yellow and orange, to red. The color directly represents the last sentiment score, by using bright red to represent sentiment value of -1 and very bad news to bright green used for the sentiment value of $+1$ and very positive news. The x position is determined in the same manner, ordering companies from left to right according to increasing recent sentiment score.

The y position of the company's representation is determined by a deviation of the recent sentiment score from the average sentiment score for the particular company, so the sentiment reversals can be easily noticed. The companies with highest sudden increase in sentiment are singled out in upper portion of display, while those companies with highest decrease in sentiment are occupying the lower portion of the display. To make this relation more obvious, we use a transformation described by Equation 3.4 to calculate the y position of the company representation y_c , as described in [17]. With $S_r(c)$ we denote deviation of current sentiment score from the average value, as described in Section 3.3.2. The values for constants are chosen empirically, after testing the system on realistic data.

$$y_c = C_1 \arctan(C_2 S_r(c)) + C_3, \quad C_1 = \frac{1.056}{\pi}, \quad C_2 = 6, \quad C_3 = 0.5 \quad (3.4)$$

The user is offered the possibility to see the details of the selected company, like numerical values of current sentiment, average sentiment, the integral text of news story, and the historical development of stock prices for the selected company. Our goal is to offer an integrated decision support tool for the users interested in analysing influence of published news articles to market movements.

3.5 Conclusion

We provided an overview of the existing text mining systems whose main purpose is predicting equity price movements on the financial markets. Some of the systems transform the input text directly to a numerical value that we call sentiment score. Others output categorical result, which in turn can be transformed into a numerical

value, thus producing a sentiment score. As an example of such sentiment score properties, we show how it relates to some of the macroeconomic variables. It is suggested that this sentiment score can be transformed to reveal sentiment reversals and deviations from the average score for a company, and such transformed indicator relates better to future returns.

Finally we present the project FINDS as an integrated system that can be used for sentiment extraction from the financial news, comparison between different classification engines, and the representation of the sentiment data to the end user.

Fig. 3.2 Components of the FINDS systems

In the time when more than 50% of all trades on certain markets are executed by algorithms, as reported in [27], real time information on sentiment of published news can be used as a valuable additional indicator and input for these algorithms. We plan on building an interface between the FINDS system and other systems which can use sentiment score information. The structure of the complete system is illustrated in Figure 3.2. In this way we could offer traders support for their buy and sell decisions, by visually filtering important news articles, but we could also support them by letting the algorithm determine the best timing and amount of the order, so the trade is executed optimally.

References

1. W. Antweiler and F.Z. Murray. Is all that talk just noise? The information content of internet stock message boards. *The Journal of Finance*, 59(3):1259–1294, 2004.
2. C. Bozic. Finds - integrative services. In *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, pages 61–62, 10–13 May 2009.
3. C. Bozic, R. Riordan, D. Seese, and C. Weinhardt. Towards a benchmarking framework for financial text mining. In D. Thomas, J. Krämer, R. Studer, and C. Weinhardt, editors, *Information Management and Market Engineering*, volume 2 of *Studies on eOrganisation and Market Engineering*, pages 21–36. KIT Scientific Publishing, Karlsruhe, 2010.
4. C. Bozic and D. Seese. Neural networks for sentiment detection in financial text. In *Proceedings of the 14th International Business Research Conference*, Melbourne, Victoria, Australia, April 2011. World Business Institute Australia.
5. C. Bozic and D. Seese. News analytics and text sentiment visualization in finance. In *SYMOPIS, 38. Symposium on Operational Research*, pages 127–130. Faculty of Economics, University of Belgrade, Oktober 2011.
6. C. Bozic and D. Seese. News analytics: Exploring predictive power of aggregated text sentiment measure. In *Proceedings of Annual Paris Conference on Money, Economy and Management*, Melbourne, Victoria, Australia, Juli 2011. World Business Institute Australia.
7. R. Brown. *Incorporating news into algorithmic trading strategies: Increasing the signal-to-noise ratio*, volume The Handbook of News Analytics in Finance, chapter 14, pages 307–309. Wiley Finance, 2011.
8. S.K. Chalup and A. Mitschele. Kernel methods in finance. In D. Seese, C. Weinhardt, and F. Schlottmann, editors, *Handbook on Information Technology in Finance*, International Handbooks on Information Systems, pages 655–687. Springer Berlin Heidelberg, 2008.
9. V. Cho and B. Wüthrich. Combining forecasts from multiple textual data sources. In N. Zhong and L. Zhou, editors, *Methodologies for Knowledge Discovery and Data Mining*, volume 1574 of *Lecture Notes in Computer Science*, pages 174–179. Springer Berlin / Heidelberg, 1999.
10. V. Cho, B. Wüthrich, and J. Zhang. Text processing for classification. Technical report, The Hong Kong University of Science and Technology, 1998.
11. W.S.V. Cho. *Knowledge discovery from distributed and textual data*. PhD thesis, Department of Computer Science, Hong Kong University of Science and Technology, 1999.
12. M. Costantino. *Financial Information Extraction using pre-defined and user-definable Templates in the LOLITA System*. PhD thesis, Laboratory for Natural Language Engineering, Department of Computer Science, University of Durham, 1997.
13. M. Costantino. The LOLITA user-definable template interface. *Journal of Computing and Information Technology - CIT*, 9(1):55–69, 2001.
14. M. Costantino, R.J. Collingham, and R.G. Morgan. Natural language processing in finance: Automatic extraction of information from financial news articles. Laboratory for Natural Language Engineering, Department of Computer Science, University of Durham, UK, 1995.
15. M. Costantino, R.J. Collingham, and R.G. Morgan. Financial information extraction at the university of Durham. In *Artificial Intelligence in Accounting & Auditing: International Perspectives*. Markus Wiener Publishers, 2005. exists also in Proceedings of the II Meeting of Artificial Intelligence in Accounting, Finance and Tax, University of Huelva, Spain, September 1996.
16. M. Costantino, R.G. Morgan, R.J. Collingham, and R. Carigliano. Natural language processing and information extraction: Qualitative analysis of financial news articles. In *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, pages 116–122.
17. A. Darmoul, G. Hackensellner, and P. Rouast. Findsvisions. Abschlusspräsentation, 07 2011. Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany.
18. S.R. Das and M.Y. Chen. Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388, 2007.

19. P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.
20. J. Doornik, M. Arellano, and S. Bond. Panel data estimation using DPD for OX. Nuffield College, Oxford, 2001.
21. T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006.
22. T. Fawcett and F. Provost. Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 53–62, New York, NY, USA, 1999. ACM.
23. G. Fung, J. Yu, and W. Lam. News sensitive stock trend prediction. In M.-S. Chen, P. Yu, and B. Liu, editors, *Advances in Knowledge Discovery and Data Mining*, volume 2336 of *Lecture Notes in Computer Science*, pages 481–493. Springer Berlin / Heidelberg, 2002.
24. G.P.C. Fung, J. Xu Yu, and L. Wai. Stock prediction: Integrating text mining approach using real-time news. In *Computational Intelligence for Financial Engineering, 2003. Proceedings. 2003 IEEE International Conference on*, pages 395 – 402, march 2003.
25. G. Gidófalvi and C. Elkan. Using news articles to predict stock price movements, 2003. Draft Tech Report, Department of Computer Science and Engineering, University of California, San Diego, USA.
26. G. Gyozo. Using news articles to predict stock price movements, 2001. Project Report, Department of Computer Science and Engineering, University of California, San Diego, USA.
27. T. Hendershott and R. Riordan. Algorithmic trading and information. Working Papers 09-08, NET Institute, March 2009. University of California, Berkeley, USA.
28. S. Hitzemann, M. Uhrig-Homburg, and K-M. Ehrhart. The impact of the yearly emissions announcement on CO2 prices: An event study. In T. Dreier, J. Krämer, R. Studer, and C. Weinhardt, editors, *Information Management and Market Engineering*, volume 2 of *Studies on eOrganisation and Market Engineering*, pages 63–78. KIT Scientific Publishing, Karlsruhe, 2010.
29. International Monetary Fund. World economic outlook database, 2011. <http://www.imf.org/external/ns/cs.aspx?id=28>.
30. Thorsten J. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.
31. John Kittrell. *Sentiment Reversals as Buy Signals*, volume The Handbook of News Analytics in Finance, chapter 9, pages 231–244. Wiley Finance, 2011.
32. V. Lavrenko, D. Lawrie, P. Ogilvie, and M. Schmill. Electronic analyst of stock behavior. Project Draft, 1999. Center for Intelligent Information Retrieval, University of Massachusetts Amherst, USA.
33. V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan. Language models for financial news recommendation. In *Proceedings of the ninth international conference on Information and knowledge management*, CIKM '00, pages 389–396, New York, NY, USA, 2000. ACM.
34. V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan. Mining of concurrent text and time-series. In *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.
35. S.K.F. Leung. Automatic stock market predictions from world wide web data. Master's thesis, Department of Computer Science, Hong Kong University of Science and Technology, 1997.
36. X. Liang. Impacts of internet stock news on stock markets based on neural networks. In J. Wang, X.F. Liao, and Z. Yi, editors, *Advances in Neural Networks ? ISNN 2005*, volume 3497 of *Lecture Notes in Computer Science*, pages 811–811. Springer Berlin / Heidelberg, 2005.
37. X. Liang and R.C. Chen. Mining stock news in Cyberworld based on natural language processing and neural networks. In *Neural Networks and Brain, 2005. ICNN B '05. International Conference on*, volume 2, pages 893 –898, oct. 2005.
38. U. Lösch. Event and sentiment detection in financial markets. In *Ph.D. Symposium at the 5th European Semantic Web Conference*, Juni 2008.

39. U. Lösch, R. Studer, and C. Weinhardt. New event detection in financial news analysis. In T. Dreier, J. Krämer, R. Studer, and C. Weinhardt, editors, *Information Management and Market Engineering*, volume 2 of *Studies on eOrganisation and Market Engineering*, pages 3–20. KIT Scientific Publishing, Karlsruhe, 2010.
40. M.-A. Mittermayer. Forecasting intraday stock price trends with text mining techniques. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, jan 2004.
41. M.-A. Mittermayer. *Einsatz von Text Mining zur Prognose kurzfristiger Trends von Aktienkursen nach der Publikation von Unternehmensnachrichten*. PhD thesis, Uniersitt Bern, 2006.
42. M.-A. Mittermayer and G.F. Knolmayer. Newscats: A news categorization and trading system. *Data Mining, IEEE International Conference on*, pages 1002–1007, 2006.
43. R. Morgan, R. Garigliano, P. Callaghan, S. Poria, M. Smith, and C. Cooper. University of Durham: description of the LOLITA system as used in MUC-6. In *Proceedings of the 6th conference on Message understanding*, MUC6 '95, pages 71–85, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.
44. B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2:1–135, January 2008.
45. D. Peramunetilleke and R.K. Wong. Currency exchange rate forecasting from news headlines. In *ADC '02: Proceedings of the 13th Australasian database conference*, pages 131–139, Darlinghurst, Australia, Australia, 2002. Australian Computer Society, Inc.
46. J.C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C.J.C. Burges, A.J. Smola, editors, *Advances in Kernel Methods*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.
47. F.J. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 445–453, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
48. R. Riordan, A. Storkenmaier, and M. Wagener. Public Information Arrival: Price Discovery and Liquidity in Electronic Limit Order Markets, 2011. Working Paper Series, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany.
49. A. Schulz, M. Spiliopoulou, and K. Winkler. Kursrelevanzprognose von ad-hoc-meldungen: Text mining wider die informationsüberlastung im mobile banking. *Wirtschaftsinformatik*, 2:181–200, 2003.
50. R. Schumaker and H. Chen. Textual analysis of stock market prediction using financial news articles. In *12th Americas Conference on Information Systems (AMCIS-2006)*, 2006.
51. R.P. Schumaker. Analyzing representational schemes of financial news articles. In *The Third China Summer Workshop on Information Systems*, 2009.
52. R.P. Schumaker. An analysis of verbs in financial news articles and their impact on stock price. In *NAACL Workshop on Social Media and Computational Linguistics*, 2010.
53. R.P. Schumaker. Analyzing parts of speech and their impact on stock price. In *21st Annual Conference of the International Information Management Association*, 2010.
54. R.P. Schumaker. Analyzing parts of speech and their impact on stock price. *Communications of the International Information Management Association*, 10:1–10, 2010.
55. R.P. Schumaker and H. Chen. A quantitative stock prediction system based on financial news. *Information Processing and Management*, 45:571–583, 2009.
56. R.P. Schumaker and H. Chen. Textual analysis of stock market prediction using breaking financial news: The AZFINTEXT system. *Association for Computing Machinery Transactions on Information Systems*, 27(2):12:1–12:19, 2009.
57. R.P. Schumaker and H. Chen. A discrete stock price prediction engine based on financial news. *IEEE Computer*, 43:51–56, 2010.
58. R.P. Schumaker, Y. Zhang, and C.-N. Huang. Sentiment analysis of financial news articles. In *20th Annual Conference of International Information Management Association*, 2009.

59. A. Šilić and B. Bašić. Visualization of text streams: A survey. In R. Setchi, I. Jordanov, R. Howlett, and L. Jain, editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6277 of *Lecture Notes in Computer Science*, pages 31–43. Springer Berlin / Heidelberg, 2010.
60. A. Storkenmaier, M. Wagener, and C. Weinhardt. Public Information in Fragmented Markets, 2011. Working Paper Series, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany.
61. P.C. Tetlock. Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, 62(3):1139–1168, 2007.
62. P.C. Tetlock, M. Saar-Tsechansky, and S. Macskassy. More than words: Quantifying language to measure firms’ fundamentals. *Journal of Finance*, 63(3):1437–1467, 2008.
63. J.D. Thomas. *News and trading rules*. PhD thesis, Carnegie Mellon University, 2003.
64. K.M. Tolle and H. Chen. Comparing noun phrasing techniques for use with medical digital library tools. *Journal of the American Society for Information Science*, 51(4):352–370, 2000.
65. C. Ullrich, D. Seese, and S. Chalup. Foreign exchange trading with support vector machines. In R. Decker and H.-J. Lenz, editors, *Advances in Data Analysis*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 539–546. Springer Berlin Heidelberg, 2007.
66. B. Wüthrich, D. Permunetilleke, S. Leung, V. Cho, J. Zhang, and W. Lam. Daily prediction of major stock indices from textual www data. In *Knowledge Discovery and Data Mining - KDD-98*, pages 364–368, 1998.
67. S.S. Zhang, C. Weinhardt, and R. Riordan. Market responses to the introduction of interactive data: The case of XBRL. In T. Dreier, J. Krämer, R. Studer, and C. Weinhardt, editors, *Information Management and Market Engineering*, volume 2 of *Studies on eOrganisation and Market Engineering*, pages 55–62. KIT Scientific Publishing, Karlsruhe, 2010.