Forecasting Intraday Stock Price Trends with Text Mining Techniques*

Marc-André Mittermayer
University of Bern, Institute of Information Systems
mittermayer@iwi.unibe.ch

Abstract

In this paper we describe NewsCATS (News Categorization and Trading System), a system implemented to predict stock price trends for the time immediately after the publication of press releases. NewsCATS consists mainly of three components. The first component retrieves relevant information from press releases through the application of text preprocessing techniques. The second component sorts the press releases into predefined categories. Finally, appropriate trading strategies are derived by the third component by means of the earlier categorization.

The findings indicate that a categorization of press releases is able to provide additional information that can be used to forecast stock price trends, but that an adequate trading strategy is essential for the results of the categorization to be fully exploited.

1. Introduction

Stock price trend forecasting based on structured data enjoys great popularity. Numerous publications describe data mining applications that try to predict the immediate future of stock prices or indices [1][2][3]. However, approaches that deal with unstructured data (i.e., text mining approaches) are hardly ever used owing to the difficulty involved in extracting the relevant information with these. Forecasting techniques that rely on structured information disregard the fact that the expectations of traders are built up to a certain extent from unstructured information.

U.S. companies are required by the Securities Exchange Act 1934 to guarantee simultaneous public disclosure of "material non-public information" because of the potential importance of such events for investors. This information includes earning figures, acquisitions and divestitures of businesses, and retirements from the Board of Directors. Nearly all companies in the U.S. have this information published as press releases through external partners to ensure compliance with the legal requirements. PRNewswire and Businesswire are at the hub in the publication of such press releases. Between them, they control about 99% of the market, each being responsible for approximately half of all press releases. Press releases are one good source of information for traders, because they may reveal unexpected information and thus have a high capability to move stock prices abruptly. Information not falling under the Securities Exchange Act 1934 can be published in "conventional" news articles or through other channels.

Negative press releases, such as bad earnings reports, normally cause traders to sell stocks, which translates into a decline in the stock price. By analogy, traders tend to buy stocks after such positive press releases as good earnings reports. This translates into buying pressure and increases the stock price. Moreover, the effect of new information on the stock price is also heavily dependent on expectations (e.g., Consensus Estimates, "Whisper Numbers"). Unfortunately, it turns out that trading strategies derived from the difference between the expected and the real numbers often do not work out.

This provides justification for the model most frequently used to describe the changes in stock prices, the random walk. It is now widely accepted that the random walk, despite its simplicity, is one of the best models for forecasting stock prices. However, it has been shown elsewhere [4] that the random walk model might not be appropriate for the description of intraday stock prices.

In this paper we assume that the probabilities of the paths in the random walk model are not the same immediately after a press release and

* This paper has benefited from discussions with and comments from Marc Heissenbüttel (University of Bern, Institute of Computer Science and Applied Mathematics).

that this skewness can be derived solely from the content of the press releases, with no account taken of expectations or other information. If this assumption holds, it should be possible to train a system by means of a supervised learning algorithm which is able to detect and exploit these facts. This was the motivation behind the implementation of a system called NewsCATS (News Categorization and Trading System), which automatically analyzes and categorizes press releases and derives stock trading recommendations from them. NewsCATS differs from previously developed systems mainly in the way the learning examples are chosen and in the determination of the "best" trading strategy. NewsCATS was tested on press releases and stock price data from 2002. The results indicate that NewsCATS can provide trading strategies which significantly outperform a trader buying and shorting stocks randomly immediately after the publication of a press release.

The rest of this paper is organized as follows. The next section gives an overview of related work in the fields of text mining and stock price trend forecasting from unstructured data. In terms of text mining, we focus especially on text preprocessing and automatic text categorization, since these are the techniques used in the work that has culminated in this paper. In Section 3 we introduce NewsCATS by briefly discussing its architecture and implementation. The performance of NewsCATS is then evaluated in Section 4. Section 5 summarizes the main findings.

2. Related work

2.1. Text preprocessing and automatic text categorization

Most algorithms used in automatic text categorization (ATC) are familiar from data mining applications. The data analyzed by data mining are numeric, which means they are already in the format required by the algorithms. These algorithms can be applied in ATC, but first it is necessary to convert the content of the documents to a numeric representation. This step is called text preprocessing, and it is often divided into the activities feature extraction, feature selection, and document representation [5].

Feature extraction is the first step in text preprocessing and consists mainly in parsing the document collection. The goal is to generate a dictionary of words and phrases (i.e., features) that describes the document collection adequately. It is common to distinguish between local dictionaries, which means separate dictionaries for each category, and universal dictionaries, with a single dictionary for the whole document collection. The feature candidates are first compared against a list of stop words, and the dictionary is then usually free of "noise" (e.g., articles, prepositions, numbers). Furthermore, word stemming techniques can be applied so that features that differ only in the affix (suffix or prefix), i.e., words with the same stem, are treated as single features. Commonly applied word stemming techniques are affix removal, successor variety, n-grams, table lookup, peak & plateau, and Porter's algorithm [6][7].

Feature extraction is followed by feature selection. The main objective of this phase is to eliminate those features that provide few or less important items of information. Indicators commonly used to determine feature importance are term frequency (TF), inverse document frequency (IDF), and their product (TF×IDF). When TF is used it is assumed that important terms occur in the document collection more often than unimportant ones. The application of IDF presupposes that the rarest terms in the document collection have the highest explanatory power. With the combined procedure TF×IDF the two measures are aggregated into one variable. Whatever metric is used, at the end of the feature selection process only the top n words with the highest scores are selected as features. While more sophisticated feature selection techniques, such as information gain, Chi-square, correlation coefficient, and relevance score, have been proposed, the above techniques (especially TF) have proved very efficient [8].

Document representation is the final task in text preprocessing. At this stage the documents are represented in terms of the features to which the dictionary has been reduced in the preceding steps. Thus, the representation of a document is a feature vector of n elements, where n is the number of features remaining when the selection process is complete. The whole document collection can therefore be seen as an m×n-feature matrix F (with m as the number of documents), where the element f_{ij} represents the frequency of occurrence of feature j in document i. Typical frequency measures are, again, TF, IDF, and TF×IDF, but a difference from the previous task is that these frequencies are now measured per document. Sometimes the frequency measure is limited to the values $\{0, 1\}$, which indicate whether or not a certain feature appears at all in the document (binary representation). At the end, the feature vectors are usually cosine normalized, since some of the ATC classifiers require feature vectors of length 1 [9].

In recent years, various techniques have been developed to reduce the size of the feature matrix F, which is sometimes enormous. These techniques rely primarily on the assumption that a large number of features are close to being synonymous. Examples of these techniques are term clustering and latent semantic indexing [10].

Major approaches for ATC classifiers involve the use of decision trees, decision rules, k-nearest neighbors, Bayesian approaches, neural networks, regression-based methods, and vector-based methods. Descriptions of these algorithms can be found elsewhere (e.g., [5] and [11]).

At this point, only one representative of the vector-based methods, called "Support Vector Machines" (SVM), is briefly discussed, because NewsCATS is based on this classifier. The difference between SVM, first introduced by Cortes and Vapnik [12], and the other classifiers mentioned above is that in addition to positive training documents, SVM also needs a certain number of negative training documents which are untypical for the category considered. SVM then searches for the decision surface that best separates the positive from the negative examples in the n-dimensional space (determined by the n features). The document representatives closest to the decision surface are called support vectors. The result of the algorithm remains unchanged if documents that are not support vectors are removed from the set of training data.

An advantage of SVM is its superior runtime behavior during the categorization of new documents: only one dot product per new document has to be computed. A disadvantage is the fact that a document could be assigned to several categories because the similarity is typically calculated individually for each category. Nevertheless, SVM is a very powerful method and has outperformed others in several studies [11][13][14][15][16].

2.2. Stock price trend forecasting using unstructured data

Wüthrich et al. [17], in 1998, analyzed news articles, collected from five popular financial websites, available before the opening of the Hong Kong stock market with several text mining techniques (k-nearest-neighbor and different types of neural networks). This analysis led to a forecast of whether the Hang Seng would go up (more than 0.5%), go down (more than 0.5%), or remain steady (between 0.5% and –0.5%) in the upcoming trading session. An average accuracy of 46% was obtained, which is significantly better than the accuracy of a random predictor, which would achieve no more than 33% accuracy.

The special feature of this work is the use of a priori domain knowledge. A dictionary consisting of 392 keywords, each considered a typical buzzword capable of influencing the stock market in either direction, was defined by several experts. Further focuses of the paper included daily data (close-to-close returns) and information available hours before the opening of the stock market. With their significant results the authors provide evidence against the Efficient Market Hypothesis [18], which states that new information is usually incorporated into stock prices within a *very* short time.

Another approach to stock price trend forecasting, one that entails correlation of the content of news articles with trends in financial time series, is described elsewhere [19]. The focus there is on intraday stock prices available at 10-minute intervals, and a priori domain knowledge is not taken into account. The authors measured the performance of their system by carrying out a market simulation. Their trading policy was to take profits of 1% or more immediately or to wait for 60 minutes and take a loss if necessary. This strategy led to an average profit per trade of 0.23%.

The same data were reused subsequently [20] to determine, among other things, the best duration of the holding period. According to the findings, the purchases or short sales should generally be evened up after 20 minutes. However, no market simulation was performed to confirm these results.

3. Concept of NewsCATS

3.1. Architecture

In this section, the architecture of NewsCATS (News Categorization and Trading System) is described. NewsCATS is designed to

1. automatically preprocess incoming press releases.
2. categorize them into different news types.
3. derive trading rules for the corresponding stock.

NewsCATS provides an engine for each of these tasks: the Document Preprocessing Engine, the Categorization Engine, and the Trading

Engine. Figure 1 gives an overview of the high-level architecture of NewsCATS.

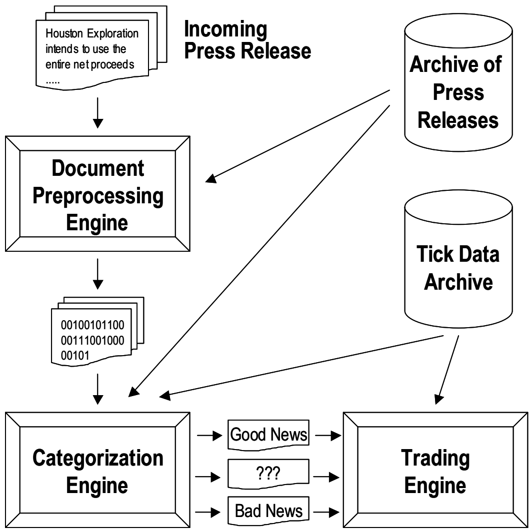

Figure 1. Architecture of NewsCATS

NewsCATS is connected to an archive of press releases and to an archive of intraday trades and quotes. With these archives NewsCATS is able to learn a set of categorization rules that allow the Categorization Engine to sort new press releases automatically into a defined number of categories. Each of these categories is associated with a specific impact on the stock prices, e.g., increase or decrease.

Depending on the results yielded by the Categorization Engine (i.e., the category assigned to the new press release) the Trading Engine produces trading signals that can be executed via an online broker or other intermediaries.

3.2. Implementation

The Document Preprocessing Engine of NewsCATS is implemented with JAVA as the programming language. During the feature extraction phase the engine is able to select from various stemming algorithms (table lookup, peak & plateau, and Porter's Algorithm) and to remove predefined stop words. Feature selection is performed by choosing TF, IDF, or TF×IDF as the measure of frequency. Document representation can be performed with a boolean measure of frequency or with TF, IDF, or TF×IDF. The Document Preprocessing Engine is further able to create local dictionaries if required. The output is forwarded to the Categorization Engine, which consists of the categorization component of the SVM Light Classifier [21]. The host application is written in Visual BASIC and also contains the Trading Engine. On arrival of a new press release, the host application launches Document Preprocessing and the Categorization Engine in that order and generates appropriate trading signals depending on their outcome.

For now, the tick-by-tick data archive consists of all historical intraday prices (trades) and bid/ask records (quotes) on all stocks in the National Market System (i.e., NYSE, NASDAQ-AMEX, and 5 regional stock exchanges) from 2002-01-01 to 2002-12-31. The archive also contains pre- and post-market trades for NASDAQ stocks.

The archive of press releases currently covers all press releases published by PRNewswire in 2002 (the press releases issued by Businesswire will be available soon). Both archives together have a total volume of approximately 150 GB, comprised of around 1 billion trades, 3 billion quotes, and 150,000 press releases. The archive is continuously extended with data from 2003.

We focus on press releases rather than on news articles in general, because we assume that due to the Securities Exchange Act 1934 press releases are the better source of unexpected information (cf. Section 1). However, we plan to extend NewsCATS to other news sources in addition, and specifically to the editorial newswires Reuters and Dow Jones.

4. Testing NewsCATS

4.1. Data

NewsCATS is being tested on a limited number of press releases. We specifically exclude all press releases that

- have no ticker symbol.
- have two or more ticker symbols.
- make no reference to the stock exchange the company is listed on.
- make reference to a stock exchange other than NYSE or NASDAQ-AMEX.
- have no subject code.

Press releases with two or more ticker symbols are excluded because determination of the publishing company turned out to be too costly for the current prototype of NewsCATS (remember that an effect on the stock price of the publishing company is the only one of interest). The absence of any reference to the stock exchange leads to exclusion because, at present, NewsCATS needs this information to gather the (historical) stock prices. Future versions of News-CATS will be able to process such press releases by looking up the stock exchange in a separate list.

We restrict the data set further by excluding all press releases of companies that have a turnover of less than US $ 5,000,000 a day (averaged over 200 randomly selected trading days in 2002), since it can be assumed that such stocks are not liquid enough to be tradable whenever required. Moreover, all press releases published before 9:30 a.m. ET or after 3:00 p.m. ET are excluded if they are NYSE listed, as are all those published before 8:00 a.m. ET or after 5:00 p.m. ET if NASDAQ listed. These restrictions arise from the tape hours of the tick-by-tick data provider and from our requirement (see Section 4.2.) for at least 60 minutes of tick-by-tick data after the publication of a press release. These constraints limit the total number of press releases used in the test to 6,602.

The stock price return accrued during the 60 minutes immediately before the publication of these 6,602 press releases is –0.01% on average. The average stock price return accrued during the 60 minutes after the publication is –0.02%. However, the standard deviation is not the same for the 60 minutes before and the 60 minutes after publication: it is 1.49% for the hour before and 2.67% for the 60 minutes immediately after. This significant difference indicates that our separation does indeed leave us with press releases that have the capability to influence stock prices (regardless of the direction).

4.2. Settings

We create three categories of press releases: "Good News", "Bad News", and "No Movers." In order to train the Categorization Engine with accurate examples for each category, we define as good news all press releases that lead the stock price concerned to peak, with an increase of at least +3%, at some point during the 60 minutes immediately after publication and have an average price level in this period that is at least 1% above the price at the time of the release. The exact values are chosen arbitrarily, but their approximate levels are based on the following reflections. The first requirement is to identify those press releases that have an immediate strong (positive) impact on the stock price, raising it by, say, +3% during the first 60 minutes. The second ensures that this effect does not hold only for a few trades, but that the press release provokes a shift of the average stock price by, say, +1% that persists for at least an hour after its publication. Sudden short-lasting price shocks, which can be caused, for example by interventions from market makers, can usually be observed during times of low activity or during pre- and post-market hours and should be eliminated from the training process. Since we operate with very short time intervals, the Beta of a stock and, mostly, the simultaneous fluctuations of the stock market (or an industry) are irrelevant.

On the other hand, all press releases leading to a maximum price drop of 3% and an average price level 1% below the price at the time of the release are considered bad news. This separation leads to classification of 347 press releases as good news and 357, as bad news. The other 5,898 press releases are labeled "no movers."

Several classifiers encounter problems when the categories in the training set vary significantly in frequency. In such cases there may be a bias towards prediction of the more common categories, leading to a worse category performance for the rarer categories [22]. To compensate for this peculiarity, we extract exactly 200 examples from each category and use these as training examples. The remaining examples are put into a holdout set that is later used to determine the model's accuracy.

The 200 examples for each of the categories "Good News" and "Bad News" are randomly extracted from the corresponding 347 and 357 press releases. Compared with systems implemented earlier, our approach is a novel one in that the 200 training examples for the category "No Movers" are randomly chosen from a subset only. This subset consists of those 1,166 (out of the 5,898) press releases that precede, simultaneously,

- the lowest maximum price change
- the highest number of price changes

of the corresponding stock during the 60 minutes following their publication. These restrictions make sure that the only press releases included are those that are not followed by large price changes or high volatility. In this way, we artificially create high selectivity between the three categories. The remaining

$$5{,}898 - 1{,}166 = 4{,}732$$

press releases in the "No Movers" category are never used for training purposes.

The preprocessing of the press releases proceeds as follows: During the feature extraction phase we create three local dictionaries that contain words only. The entries in the dictionaries are stemmed with Porter's algorithm, meaningless stop words (especially the xml tags in the original press releases) are removed, and numbers are excluded. During feature selection we reduce each of the dictionaries to the 1,000 most meaningful terms. "Most meaningful" is used to describe those terms that reach the highest TF×IDF value. Finally, the document representation is accomplished with a boolean measure of frequency, and the feature vectors are cosine normalized for further use. Learning of the support vectors is achieved by means of the learning component of the SVM Light Classifier [21].

4.3. Output of NewsCATS

Learning of the support vectors is conducted 50 times. Each time, the

$$147 + 157 + 5{,}698 = 6{,}002$$

examples remaining when the training process is complete (i.e., the holdout set) are categorized to determine the model's accuracy. Descriptive statistics for the precision and recall measures achieved in the 50 runs are shown in Table 1. Precision is the ratio of the number of relevant documents that have been categorized into the category under scrutiny to the total number of documents that have been categorized (relevant and non-relevant documents). Recall is the ratio of the number of relevant documents that have been categorized into the category under scrutiny to the total number of relevant documents that should have been categorized.

Table 1. Precision and recall measures of 50 categorization runs

	Good News (N=147)		*No Movers (N=5,698)*		*Bad News (N=157)*		*Overall (Weighted Recall)*
	Prec.	*Rec.*	*Prec.*	*Rec.*	*Prec.*	*Rec.*	
Avg.	6%	43%	98%	59%	5%	47%	58%
Min.	5%	37%	98%	54%	4%	38%	54%
Max.	7%	50%	98%	61%	5%	54%	60%
StDev.	0%	4%	0%	2%	1%	5%	2%

In the case where the algorithm is unable to detect patterns in the training documents, the average recall for each category is equal to 33%. In our example, all categories have an average recall which is significantly above this value. The overall accuracy of the categorization (measured as the weighted recall) is almost equal to the recall of the "No Movers" category, since the vast majority of the press releases belong in this category.

The average precision of the categories "Good News" and "Bad News" is fairly low, at 6% and 5%, respectively. However, we have to consider that the precision metrics do not take proper account of how "wrong" a categorization of a press release in fact is. For instance, a press release with an impact of +2.9% on the underlying stock price probably consists of information nearly as good as that in a press release leading to a price increase of 3.1% and might therefore also be categorized as "Good News," which is unfavorable for the precision metric but theoretically correct.

The precision and recall figures for the "No Movers" category indicate that this category is characterized by high selectivity. This is further supported by a look at the category clusters of the training set formed in the feature space. Figure 2 shows the category clusters in a reduced 3-dimensional feature space, where the documents in the "No Movers" category (black spots) are pooled extremely well in one "corner" and the other categories are spread over the remaining feature space.

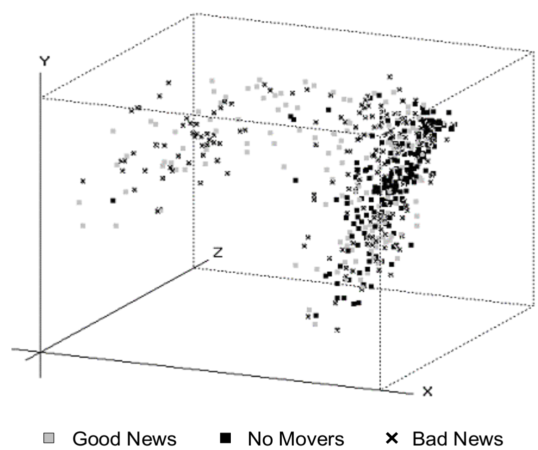

Figure 2. Feature space of training set

The selectivity of the two other categories is fairly poor. One possible explanation for this is that press releases containing good and press releases containing bad information in fact draw on a different vocabulary than the "no movers," but this treasury of words differs only slightly between the two. Consider the example of a company that is suddenly threatened with delisting (truly bad news). The corresponding press release might therefore contain something like, "Company X will be delisted from

NASDAQ." Now, let us assume that a few days later the company is no longer under threat of such delisting and publishes a press release containing the same passage except that the word "not" is inserted before "be delisted". Since the vocabularies of the good press release and the bad press release (published a few days before) are the same, the only difference is manifest in the negation, but precisely this word is probably an element in a stop word list. One possible way of tackling this problem might be to change from word-based to phrase-based preprocessing, and this is currently under investigation.

It is also interesting to note that even though the algorithm was trained with 200 examples per category, i.e., with a uniform prior distribution, it correctly sorts most of the test examples into the "No Movers" category.

After the automatic categorization the output is forwarded to the Trading Engine. This engine translates the categorization outcomes into trading signals of the types "Buy Stock," "Short Stock," and "Do Nothing." While more sophisticated trading signals are in development (e.g., including the "best" duration of the holding period), we limit our current work to these basic trading signals. Preliminary results (not shown) concerning the "best" duration of the holding period have revealed, surprisingly, that it might be better to choose a long (short) holding period for stocks with a high (low) daily turnover. Table 2 summarizes descriptive statistics for the trading recommendations generated by the Trading Engine.

Table 2. Buy and short recommendations generated by the Trading Engine

	Buy Recommendations	*Short Recommendations*
Avg.	1,330 (22%)	1,272 (21%)
Min.	1,158 (19%)	997 (17%)
Max.	1,581 (26%)	1,409 (23%)
StDev.	110 (2%)	101 (2%)

Although only 147 (157) press releases of the holdout set are primarily labeled "Good News" ("Bad News"), on average the Trading Engine recommends buying the corresponding stocks 1330 (i.e., 22% of 6,002) times and shorting 1272 times (i.e., 21%). Thus, many of the originally "no movers" are categorized into wrong categories, but reconsideration of the above example of a press release with an impact of slightly less than +3% on the underlying stock price suggests that this false categorization might turn out to be advantageous, because the stock is acquired nevertheless.

The total number of trades suggested is an average of 2,602 (43%). A system that is unable to detect patterns in the training documents would release approximately 4,000 (two-thirds) buy and short recommendations, since we performed the training with a uniform prior distribution.

4.4. Market simulation

To evaluate the performance of NewsCATS we execute the buy and short recommendations virtually using the tick-by-tick data archive of 2002. We assume that stocks can be bought or shorted exactly 2 minutes after the publication of a press release. A delay of even 2 minutes seems adequate, because the PRNewswire feed is available in real time at http://biz.yahoo.com/prnews/ and the categorization of a press release takes an average of 30 seconds. The holding period is set at 58 minutes; thus, we even up exactly 60 minutes after publication of the press release. Table 3 displays descriptive statistics for the executed trades. Since it is common to compare the performance of a trading system with a random strategy that leads to approximately the same numbers of purchases and short sales, the columns on the right in Table 3 present the results of such a "best random trader."

Table 3. Trades executed and average profit per trade

	NewsCATS		*Random Trader*	
	Trades Executed	*Avg. Profit per Trade*	*Trades Executed*	*Avg. Profit per Trade*
Avg.	2,602	0.11%	2,599	0.00%
Min.	2,477	0.03%	2,475	-0.05%
Max.	2,864	0.18%	2,860	0.06%
StDev.	96	0.06%	96	0.03%

The average profit per trade is 0.11% and 0.00%, respectively, for NewsCATS and the random trader. The number of the random trader is not significantly greater than the average stock price return during the 60 minutes after the publication of press releases (cf. Section 4.1.). On the other hand, the profit achieved by NewsCATS is significantly greater at the 1% level. This result strongly supports the assumption made in Section 1 that the probabilities of the paths in the random walk model are not the same immediately after a press release and that this skewness can be derived solely from the content of the press release.

To further improve the results, the Trading Engine determines price barriers that, if exceeded, cause the long and short positions to be evened up (even if the 58 minutes are not at an end). This means that as soon as we can take a profit (loss) of d% within the interval of]2, 60[after a press release becomes public, we do so. Otherwise, we wait until the end of the hour and take a loss (profit) if necessary. Other rules for the Trading Engine are in development, as mentioned in Section 4.3. Table 4 shows the results for the same 50 runs as are summarized in Table 3, depending on various barriers.

ances. Thus, these scenarios do not need to be considered further.

The significant differences of profits per trade (compared to the "base case") achieved by applying various upper and lower barriers can be explained by the fact that intraday price movements do not follow a random walk model [4] but are the result of the interaction between a random walk and temporary thresholds produced by limit orders [23]. If barriers are chosen appropriately this fact enables the generation of small profits even if stocks are bought and shorted completely randomly (as done by the random trader). A more detailed discussion on the different profits is, however, beyond the scope of this paper.

Table 4. Average profit per trade for various barriers

Symmetrical Barriers				*Asymmetrical Barriers: upper > |lower|*				*Asymmetrical Barriers: upper < |lower|*			
Upper Barrier	*Lower Barrier*	*News CATS*	*Random Trader*	*Upper Barrier*	*Lower Barrier*	*News CATS*	*Random Trader*	*Upper Barrier*	*Lower Barrier*	*News CATS*	*Random Trader*
infinite	*infinite*	0.11%	0.00%	*infinite*	*-3.0%*	0.05%	-0.03%	*3.0%*	*infinite*	0.15%	0.03%
3.0%	*-3.0%*	0.09%	0.00%	*infinite*	*-2.0%*	-0.01%	-0.06%	*2.0%*	*infinite*	0.17%	0.06%
2.0%	*-2.0%*	0.07%	0.00%	*infinite*	*-1.0%*	-0.05%	-0.08%	*1.0%*	*infinite*	0.21%	0.07%
1.5%	*-1.5%*	0.06%	-0.01%	*infinite*	*-0.5%*	-0.05%	-0.07%	*0.5%*	*infinite*	0.19%	0.06%
1.0%	*-1.0%*	0.05%	-0.01%	*3.0%*	*-1.0%*	-0.01%	-0.05%	*1.0%*	*-3.0%*	0.15%	0.04%
0.5%	*-0.5%*	0.05%	-0.01%	*3.0%*	*-0.5%*	-0.02%	-0.05%	*0.5%*	*-3.0%*	0.15%	0.04%
0.2%	*-0.2%*	0.04%	-0.01%	*1.0%*	*-0.5%*	0.04%	-0.01%	*0.5%*	*-1.0%*	0.06%	-0.01%

The numbers indicate that NewsCATS always outperforms the random trader. With symmetrical barriers, for instance, the profit per trade is from 0.05% to 0.11% higher. (Please note that the "base case" infinite/infinite is the case without barriers shown in Table 3.)

With asymmetrical barriers that cause the Trading Engine to take profits earlier than losses (+3%/infinite, +1%/–3%, etc.) NewsCATS performs even better. Depending on the barriers, the profit reaches up to 0.21% per trade (averaged over the 50 runs) and is therefore 0.14% higher than the one of the random trader. The average profit per trade of the individual runs in this "best" scenario ranges from 0.13% to 0.28% (not shown). In the model presented by Lavrenko et al. [19] a similar simulation was carried out with the same barriers as in our "best" scenario, but no explanation was given for these specific choices. In a single simulation run a profit of 0.23% per trade was achieved, which basically confirms our results.

With asymmetrical barriers that cause the Trading Engine to realize losses earlier than profits (infinite/–3%, +3%/–1%, etc.), both NewsCATS and the random trader show their worst perform-

After the discovery in Table 4 that the best scenarios have no lower barrier (cells with gray background in Table 4), it is interesting to engage in further investigation of the average profit achieved with different upper barriers. Therefore, a sensitivity analysis is conducted for values of the upper barrier between +0.02% and +2.0%. The Trading Engine is able to incorporate these findings into more detailed trading recommendations, such as

"Buy Stock X and Hold It
Until the Stock Price Hits the +d% Barrier."

Figure 3 shows the average profit per trade for various upper barriers and no lower barrier (i.e., lower barrier set to infinite). Obviously the difference between the average profit per trade of NewsCATS and the average profit per trade achieved by the random trader remains constant and statistically significant. The highest profit per trade is obtained with an upper barrier set at +0.9%, but it cannot be confirmed statistically that other barriers close to +1% really lead to lower profits.

Furthermore, NewsCATS is still able to yield profits if we take transaction costs into account.

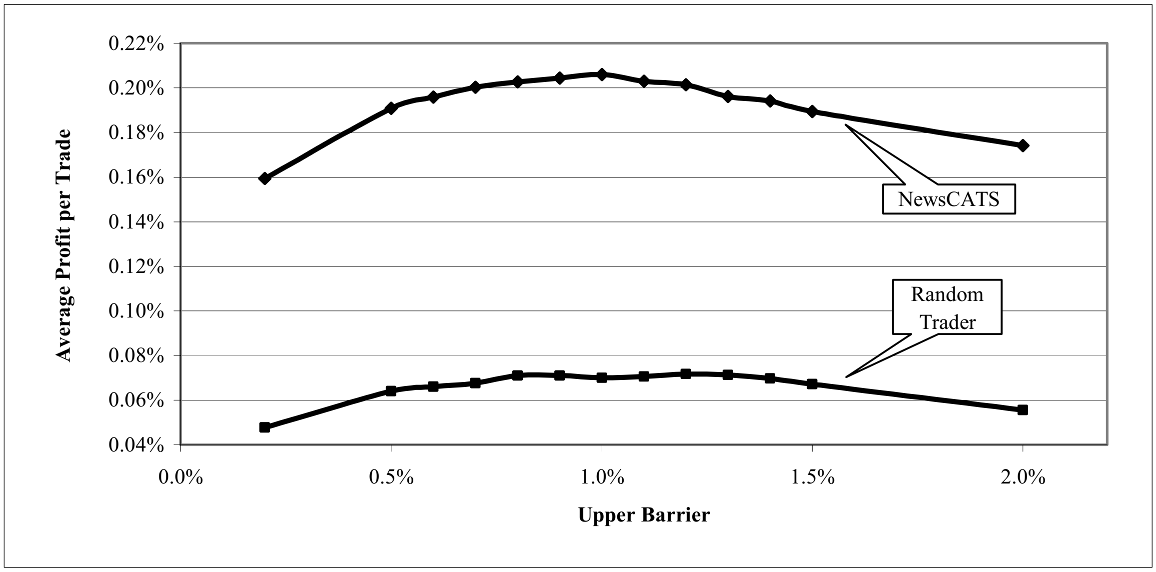

Figure 3. Average profit per trade for various upper barriers

By assuming transaction costs of US $ 10 for buying and US $ 10 for selling stocks, NewsCATS breaks even if each recommended trade is executed with an amount of

(US $ 10 + US $ 10) / 0.21% = US $ 9,524

and evened up as soon as +0.9% or more can be obtained. Since we focused the market simulation on stocks with a daily turnover of at least US $ 5,000,000, purchases or short sales above US $ 9,524 are not an obstacle for NewsCATS.

5. Summary and outlook

Based on the assumption that the random walk of stock prices immediately after the publication of a press release is skewed (which we believe can be derived solely from the content of the press release), we implemented NewsCATS, which automatically analyzes and categorizes press releases and generates stock trading recommendations. NewsCATS differs from systems developed earlier mainly in the way the learning examples are chosen and the way the trading recommendations are compiled. NewsCATS was tested on press releases and intraday stock price data from 2002. The results indicate that NewsCATS can provide trading strategies that significantly outperform a trader randomly buying and shorting stocks immediately after the publication of press releases.

However, the results also reveal that there is still much room for improvement. In particular, the output of the Categorization Engine needs to be enhanced. Since the selectivity of the "No Movers" category is good but the selectivity of the two other categories is fairly poor (as seen for instance in Figure 2), the learning could be improved by inserting a new first step to distinguish between "No Movers" and "Movers" only. In a second step the "Movers" could then be split into "Good News" and "Bad News."

Furthermore, the outcome of the categorization process depends heavily on the feature matrix created by the Document Preprocessing Engine. One possible way of improving the preprocessor is to apply a priori domain knowledge. This means that feature extraction and the feature selection phase become obsolete because the dictionary is predefined by experts. Such a dictionary consists of words and phrases that are generally regarded as buzzwords capable of influencing stock prices. However, the definition of such a dictionary reduces the flexibility of a system such as NewsCATS. Currently, NewsCATS works out the domain knowledge on its own (during feature extraction and feature selection) and is therefore able to account for vocabulary changes.

6. References

[1] M.-C. Chan, C.-C. Wong, W. F. Tse, B. Cheung, and G. Tang, "Artificial Intelligence in Portfolio Management", H. Yin, N. Allison, R. Freeman, J. Keane, and S. Hubbard (eds.), Intelligent Data Engineering and Automated Learning, Springer, Heidelberg, 2002, pp. 403-409.

[2] J. Roman and A. Jameel, "Backpropagation and Recurrent Neural Networks in Financial Analysis of Multiple Stock Market Returns", R. H. Sprague (ed.), Proceedings of the 29th Hawaii International Conference on System Sciences, Vol. 2, IEEE Computer Society Press, Los Alamitos, CA, 1996, p. 454.

[3] R. J. van Eyden, The Application of Neural Networks in the Forecasting of Share Prices, Finance and Technology Publishing, Haymarket, VA, 1996.

[4] V. Niederhoffer and M. Osborne, "Market making and Reversal on the Stock Exchange", Journal of the American Statistical Association 61 (1966) 316, ASA, Alexandria, VA, pp. 897-916.

[5] H. Brücher, G. Knolmayer, and M.-A. Mittermayer, "Document Classification Methods for Organizing Explicit Knowledge", Proceedings of the 3rd European Conference on Organizational Knowledge, Learning, and Capabilities, ALBA, Athens, 2002.

[6] J. Bollen, Text Operations,
URL: http://www.cs. odu.edu/~jbollen/spring03_IR/cs695_lect6.pdf [2003-03-20].

[7] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman, Boston, MA,1999.

[8] F. Sebastiani, "A Tutorial on Automated Text Categorisation", A. Amandi and A. Zunino (eds.), Proceedings of the 1st Argentinean Symposium on Artificial Intelligence, ASAI, Buenos Aires, 1999, pp. 7-35.

[9] O. Madani, ABCs of Text Categorization,
URL: http://classes.seattleu.edu/computer_science/csse470/Madani/ABCs.html [as of 2003-06-01].

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic indexing", Journal of the American Society for Information Science 41 (1990) 6, John Wiley & Sons, Hoboken, NJ, pp. 391-407.

[11] Y. Yang and X. Liu, "A Re-Examination of Text Categorization Methods", Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, ACM Press, New York, NY, 1999, pp. 42-49.

[12] C. Cortes and V. Vapnik, "Support-vector networks", Machine Learning 20 (1995) 3, Kluwer Academic Publishers, Hingham, MA, pp. 273-297.

[13] S. Dumais, J. Platt, and D. Heckerman, "Inductive Learning Algorithms and Representations for Text Categorization", Proceedings of the 7th International Conference on Information and Knowledge Management, ACM Press, New York, NY, 1998, pp. 148-155.

[14] M. A. Hearst, "Support Vector Machines", IEEE Intelligent Systems 13 (1998) 4, IEEE Educational Activities Department, Piscataway, NJ, pp. 18-21.

[15] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", C. Nédellec and C. Rouveirol (eds.), Proceedings of the 10th European Conference on Machine Learning, Springer, Heidelberg, 1998, pp. 137-142.

[16] G. Siolas and F. d'Alché-Buc, "Support Vector Machines based on a semantic kernel for text categorization", Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IEEE Computer Society Press, Los Alamitos, CA, 2000, pp. 205-209.

[17] B. Wüthrich, V. Cho, S. Leung, D. Permunetilleke, K. Sankaran, J. Zhang, and W. Lam, "Daily Stock Market Forecast from Textual Web Data", Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, IEEE Computer Society Press, Los Alamitos, CA, 1998, pp. 2720-2725.

[18] E. F. Fama, "The Behavior of Stock-Market Prices", The Journal of Business 38 (1965), The University of Chicago Press, Chicago, IL, pp. 34-105.

[19] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan, "Language Models for Financial News Recommendation", Proceedings of the 9th International Conference on Information and Knowledge Management, ACM Press, New York, NY, 2000, pp. 389-396.

[20] G. Gifdofalvi, Using News Articles to Predict Stock Price Movements,
URL: http://www.cs.ucsd.edu/users/gyozo/studies/cse254_AI/stock_price_prediction.pdf [2001-06-15].

[21] T. Joachims, SVM Light Classifier,
URL: http://svmlight.joachims.org/ [2002-07-05].

[22] S. Lawrence, I. Burns, A. D. Back, A. C. Tsoi, and C. L. Giles, "Neural Network Classification and Prior Class Probabilities", G. Orr, K.-R. Müller, and R. Caruana, (eds.), Tricks of the Trade, Springer, Heidelberg, 1998, pp. 299-314.

[23] C. W. J. Granger and O. Morgenstern, Predictability of Stock Market Prices, Heath Lexington Books, Lexington, MA, 1970.