2008

A Text Mining Approach to Support Intraday Financial Decision-Making

Sven S. Groth
E-Finance Lab Frankfurt, sgroth@wiwi.uni-frankfurt.de

Jan Muntermann
J.W. Goethe-University Frankfurt, muntermann@wiwi.uni-frankfurt.de

A Text Mining Approach to Support Intraday Financial Decision-Making

Sven S. Groth
E-Finance Lab Frankfurt
sgroth@wiwi.uni-frankfurt.de

Jan Muntermann
J.W.Goethe-University Frankfurt
muntermann@wiwi.uni-frankfurt.de

ABSTRACT

Developing forecasting models for estimating the behavior of capital markets is one of the most challenging tasks in financial decision support system research. Besides time series models, artificial neural network approaches and genetic algorithms, text mining technologies represent a promising approach to support financial decision-making. In this paper, the authors address the problem field of predicting stock price movements shortly after the publication of company-specific news. Incorporating the findings of current financial research, a two-stage text mining classification approach to forecast short term intraday stock price movements is presented.

Intraday event study analyses have detected significantly different intraday stock price reactions within news sub-classes. Therefore, a forecasting approach is presented that aims at identifying those news sub-classes for which most significant price reactions can be expected. Moreover, the advantage of local classifiers over global classifiers for the most relevant subclass identified is highlighted.

Keywords

Text mining, supervised learning, financial forecasting.

INTRODUCTION

The development of information systems that support financial decision-making is a difficult task, since financial processes are complex and data relationships are hard to model. Dhar and Stein (1997) identify four major difficulties when developing forecasting models to support financial decision-making:

- Due to high complexity of financial markets, models have to cover knowledge regarding financial relationships where the type and intensity of the relationship can hardly be understood completely.
- In the financial domain, forecasting models and system designs usually have to process and analyze data series featuring highly complex behavior.
- Voluminous economic and financial data series have to be analyzed, especially when exploring intraday market behavior. Large amounts of data have to be processed to extract comprehensible information to investors.
- Since financial markets are non-stationary over time, models and system design are required to be flexible regarding necessary adjustments.

In the following, the authors face these challenges and present an approach to evaluate intraday stock price adjustments subsequent to the publication of company announcements (so-called ad hoc disclosures). Latest empirical findings from financial research have revealed significant differences regarding the price effects within news sub-classes (Muntermann and Güttler, 2007). As some news sub-classes reveal either no or sporadic price effects only, the authors infer that building general forecasting models is not always preferable. Therefore, a two-stage text-mining classification approach is proposed.

The remainder of this paper is organized as follows: In the next section a literature review on most relevant research that addresses financial decision support with text mining technologies is presented. Afterwards, above mentioned two-stage text mining classification approach is presented. The paper concludes with a summary and proposals for future research.

TEXT MINING RESEARCH IN FINANCIAL DECSION-MAKING

The idea of applying text mining approaches to support financial decision making has been explored by researchers since the late nineties. Generally, these works are based on a collection of news publications and some sort of capital market reactions (such as price trends or volatility changes) following the publication date.

Among the first to make use of text mining technologies to address financial forecasting problems are Wuthrich, Cho, Leung, Permunetilleke, Sankaran and Zhang (1998) who observe financial news articles published on web-based news portals and estimate the price trends of major market indices on a daily basis. Based on these general market trend forecasts, they derive a trading strategy achieving long-term profits that outperform several actively managed funds.

Lavrenko, Schmill, Lawrie, Ogilvie, Jensen and Allan (2000) present a text mining approach to identify and recommend news stories that will most likely affect stock prices. In contrast, Wuthrich et al. (1998) are aiming at intraday decision support and address stock price adjustments in the first hour subsequent to news releases. In order to evaluate their forecasts, they present a simple trading simulation. The simulated cumulative profits are significantly higher compared to simulation runs working with randomly triggered investment decisions.

The work of Schulz, Spiliopoulou and Winkler (2003) addresses the problem field of identifying and forwarding highly relevant company announcements wireless to mobile devices of retail investors. Therefore, they classify these company announcements using the SAS Enterprise Miner in order to forecast the abnormal stock price behavior observed during the publication dates. With a classification error of 59%, when classifying relevant announcements, the authors conclude that company announcements contain too much noise.

Mittermayer (2004) presents a system design to predict stock price adjustments subsequent to the publication of press releases. Releases were classified as good or bad news if a significant price reaction has been observed during the 60 minutes following the publication date. An evaluation based on market simulations provides evidence that a trading strategy utilizing the classification model will result in significantly higher trading profits per trade compared to profits of noise traders placing random trades.

Fung, Yu and Lu (2005) propose a text mining-based system design that aims at realizing trading profits for a holding period of 1 to 7 days following a news release. Therefore, a forecasting model is developed on the basis of 350,000 Reuters news stories and stock price movements observed on the Hong Kong stock exchange.

With forecasting and evaluation periods of 1 hour to several days, these existing works disregard the empirical findings published by the financial research community. Intraday event study analyses that explore the speed at which securities adjust to new information provide evidence for shorter periods after which prices fully reflect information (e.g. Patell and Wolfson 1984; Woodruff and Senchak, 1988). Furthermore, employing content analysis techniques was found most promising when capital markets react promptly to new information (Robertson, Geva and Wolff, 2007). Moreover, significant differences in price effects were observed among news sub-classes (Muntermann and Güttler, 2007). Incorporating these findings, it seems most promising to forecast price trends for news classes for which significant capital market reactions have been revealed.

AN INTRADAY TEXT MINING APPROACH

Dataset

The authors make use of the dataset evaluated by Muntermann and Güttler (2007), comprising ad hoc disclosures published by the Deutsche Gesellschaft für Ad-hoc-Publizität (DGAP) on behalf of the companies whose shares are admitted to trading on an organized market in Germany. To fulfill legal requirements, these companies have to publish immediately any insider information or other information being highly relevant to investors. The dataset comprises of 160 ad hoc disclosures that were published between 2003-08-01 and 2004-08-31 during stock exchange trading hours. For each of these ad hoc disclosures, the company's intraday stock price series (of the publication date) were obtained. Consequently, the authors are able to map ad hoc disclosure contents with the intraday price behavior following its publication.

In their intraday event study analysis, Muntermann and Güttler (2007) provide six ad hoc disclosure categories and analyze the abnormal stock price behavior for each category. Table 1 illustrates their results on category-varying abnormal stock price reactions.

	Ad hoc Disclosure Category					
	Financial Statements[1]	Expected Positive Events[2]	Expected Negative Events[3]	Takeovers	Disposition of Assets	Changes in Management
Observations	61	42	36	14	14	10
Abnormal Price Reactions	continuous	non-continuous	non-continuous	non-continuous	non-continuous	none

[1] Financial statements include annual, quarterly and dividend statements.
[2] Expected positive events include takeovers of the company / of companies, substantial orders, cooperation, changes in strategy, stock repurchases, stock splits and innovations.
[3] Expected negative events include increases of share capital, bankruptcies, litigations, capital reductions and adverse value adjustments by the company.

Table 1. Dataset Description

The empirical findings provide evidence that depending on an ad hoc disclosure's category membership, there exist significant differences in the price reactions following its publication. Whereas significant price reactions can be observed continuously for a defined period following the publication of financial statements, noncontinuous or no (i.e. not significant) stock price reactions can be observed for other categories. Previous text mining analyses on company disclosures and stock price adjustments such as those presented by Schulz et al. (2003) and Mittermayer (2004) do not take into account these empirical findings.

General Setup

It is against this background that it seems valuable to be able to automatically identify the potentially significant subset of all ad hoc disclosures, namely "financial statements". Unfortunately, the distributor of ad hoc disclosures does not label the forwarded company announcements accordingly, but a classifier might be able to do so.

Pursuing this attempt, a system that assigns each released ad hoc disclosure to one of the two categories "financial statements" and "non-financial statements" is proposed. As illustrated in Figure 1, *Classifier A* will be trained to fulfil this task: After the documents (i.e. the ad disclosure texts) were labelled, these need to pass through pre-processing in order to have the right format to be further processed by the learning algorithm (classifier, Support Vector Machine (SVM)). Once the learning algorithm has created a classifier, the resulting model can be used to map arbitrary documents onto proper class labels (see Figure 2).

Moreover, given above described continuous abnormal price effects within the subgroup "financial statements", one is expected to find there a more precise ontology than on a global level. In order to test this hypothesis ($\boldsymbol{H_0}$), two more classifiers, i.e. *Classifier B* and *Classifier C* (see Figure 1), are introduced and compared with each other. Thereby both classifiers learn on ad hoc disclosures that are labelled according to their short-term price effect subsequent to publication. While *Classifier B* constitutes a global classifier, local *Classifier C* is derived from a "financial statements" sub-sample. Hypothesis $\boldsymbol{H_0}$ will be corroborated if *Classifier C* turns out to be superior to *Classifier B* with regard to usual evaluation metrics such as accuracy, recall and precision.

If $\boldsymbol{H_0}$ can be corroborated, a two-stage classification approach, such as the one shown in Figure 2, is most likely to improve forecasting ability compared to a system being solely based on *Classifier B*. Thereby within a first step, above introduced *Classifier A* is applied to determine an appropriate local classifier (here: *Classifier C*) which then conducts the final classification task.

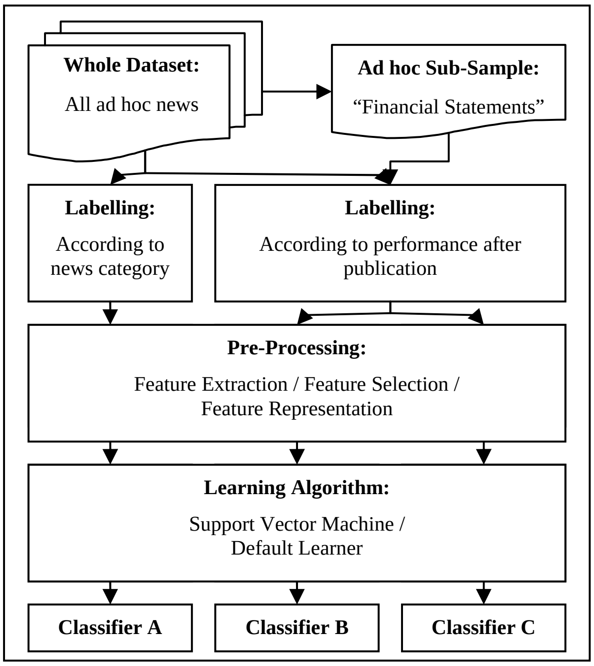

Figure 1. Learning Phase

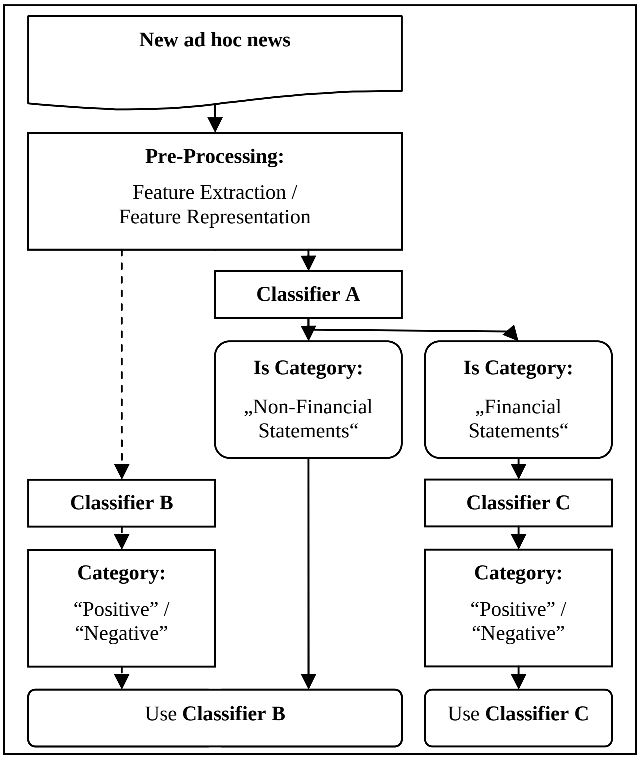

Figure 2. Application Phase

Labelling

Enabling below learning algorithms during *supervised learning* to actually know what to look for, above documents first of all need to be labelled: For *Classifier A*, all ad hoc disclosures need to be assigned to one of the categories "financial statements" (fs) and "non-financial statements" (nfs). Learning tasks *B* and *C* (*Classifier B, Classifier C)* require documents labelled "positive", "negative" or "neutral" as input.

Labelling the documents for learning task *A* (*Classifier A*) is straight forward. Leis and Nowak (2001) identified nine major categories that cover all kinds of company announcements. Adopting their (nine) category definitions, each document in the dataset was manually assigned to one of these categories. If an ad hoc disclosure turned out to cover two or more categories, it was decided in favour of the one that apparently "stood out". Afterwards, each Leis and Nowak (2001) news category was mapped with the appropriate two-class category that served as input for the learning task. The number of ad hoc disclosures belonging to the category "financial statements" is equal to the quantity identified by Muntermann and Güttler (2007).

The authors decided to apply different return metrics as a proxy for abnormal price reactions within learning task *B* and *C*. Three of them are introduced and applied in the following:

The "simple return" measure (*sr*) takes into account the underlying stock's prices p_e at event (publication) time and at event time plus an interval of 15 minutes, see Formula 1. This measure obviously neither takes into account price movements within the interval nor price movements by an index (market trend). Analogue to Mittermayer (2004), it is assumed that the interval, in our case the 15 minutes following the news publication during which Muntermann and Güttler (2007) detected abnormal price behavior, is too short to be significantly influenced by simultaneous market fluctuations. The documents are labelled "positive" if the performance measure *sr* exceeds the upper barrier b_u. Achieving a *sr* below b_u and above b_l (lower barrier) entails the label "neutral". If the upper and lower barriers are well chosen, *sr* should be capable of capturing abnormal price reactions.

$$sr = \frac{p_{e+15}}{p_e} \tag{1}$$

In contrast, the "average return" measure (*av*) also takes into account price movements within the analyzed interval. Thereby the (ln-) returns for each tick *t* within the interval are calculated and averaged at the end (see Formula 2). The documents are labelled "positive" if av_n (for news item *n*) is above the average of all av_n in the dataset (or subset); and "negative" otherwise.

$$av_n = \frac{\sum \ln\left(\frac{p_{t+1}}{p_t}\right)}{t} = \frac{\sum r_t}{t} \tag{2}$$

The third "peak" measure (*peak*) delivers (two) binary results. If return r_t crosses barriers b_u or b_l at least once during the interval, i.e. 15 minutes after publication, the document is assigned a *1*, otherwise a *0*. The documents are labelled "positive" if either one or both barriers have been crossed.

$$x_t = \begin{cases} 1, if\ r_t > b_u\ or\ r_t < b_l \\ 0, else \end{cases} \tag{3}$$

$$peak_n = \begin{cases} positive, if\ any\ x_t = 1 \\ negative, if\ all\ x_t = 0 \end{cases} \tag{4}$$

Text Pre-Processing

As "traditional" data mining learning algorithms which are also applied in text mining are not capable of coping with plain text, the main pre-processing task is to be found with the transformation of textual documents into a numeric representation. The applied pre-processing procedure is explained along with Brücher, Knolmayer and Mittermayer (2002), conducting the basic steps: *Feature Extraction*, *Feature Selection,* and *Feature Representation*.

Feature Extraction

The goal of this phase is to generate a dictionary of words and phrases that describes the document collection adequately. Thereby a tokenizer splits up the whole text into individual units, i.e. in this case: words. The applied simple *StringTokenizer* uses the Unicode specification to identify separators by non-letter characters (Wurst, 2007). It follows that – after an additional German stopword list and a threshold on the number of documents each token occurs in has been made use of – the feature set consists neither of punctuation marks or hardly to interpret numbers, nor of "noise" (i.e. words with little meaning, but frequent appearances such as articles or prepositions).

Moreover, *stemming*, i.e. mapping different grammatical forms of a word to a common stem / term by removing the affixes (suffixes or prefixes), usually reduces the amount of features within the feature set (Witten, Moffat and Bell, 1999). Given its' nature, stemming is expected to improve the learning algorithms' performance, but is also questioned (Hull, 1996). The authors apply (alternating) both the *Porter Stemmer* (1980) and a simple *GermanStemmer* (Wurst, 2007).

Further feature extraction techniques such as the substitution of synonyms or the replacement of hyponyms by a more common term were not implemented.

Feature Selection

Having extracted the tokens from the corpus, one is usually confronted with a huge feature set consisting of several thousand features. It is against this background that mainly for (computing) efficiency reasons and sometimes for accuracy reasons as well, those features that contain few or relatively less important information are eliminated. For large scale text mining problems, feature scoring methods (filter methods) are recommended (Forman, 2003). Hereby, each feature is ranked by one feature selection metric and at the end, the most highly ranked features are chosen to remain in the final feature set. Performing an empirical comparison of different feature selection methods, especially suited for SVMs, Forman (2003, p. 1298) finds that above mentioned *established* methods do not "perform better than using all features available". Therefore the authors of this study do not make use of these methods.

Feature Representation

Finally, each document is represented by previously extracted and selected number of features. The respective feature weightings may for example be generated by above mentioned frequency measures. While *tf* denotes the term frequency of a feature in a document (document dependent), *df* denotes the number of documents the feature appears in (document independent). Given Formula 5, it becomes clear that *tfidf* implies a further feature extraction component because a higher document frequency lowers its respective *tfidf* weight. *Tfidf* is used for feature representation (Lewis, 1992).

$$tfidf\,(w) = tf\,\log\left(\frac{N}{df\,(w)}\right) \qquad \textbf{(5)}$$

w = feature, N = Total number of documents

Text Mining and Classification

After pre-processing, the resulting document (row) – feature (column) representation (table) provides the basis for the following text classification. As both comparative empirical studies (Joachims, 1998; Yiming and Xin, 1999) and the authors' own pre-tests have revealed that the performance of SVMs for classification tasks seems to be superior to other methods, the major learning and prediction process is based on this classification algorithm.

SVM was firstly introduced by Vapnik (1995) for solving two-class recognition problems. In contrast to parametric classification methods, the addressed non-parametric method SVM interprets the input matrix spatially. Given the two-dimensional example space in Figure 3, the general idea is to find the decision surface that maximizes the margin between the data points (classes). In this case, the data points are linearly separable. Separation in below two-dimensional space, however, is not given by a "simple" dividing line, but by the intersection of (x,y)-space and a hyperplane, being orthogonal to (x,y)-space. Thereby maximizing the margin between a separating hyperplane and the nearest data points is undertaken by means of structural risk minimization.

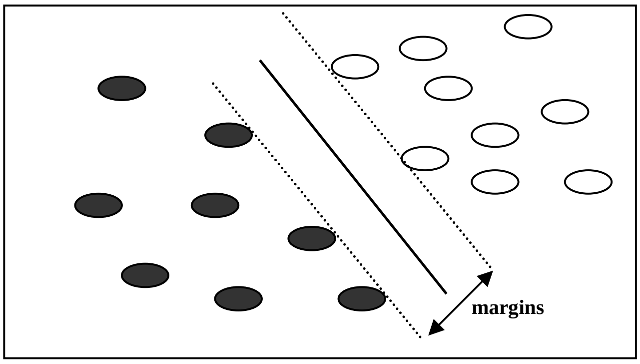

Figure 3. SVM Example

In case of originally linearly non-separable data points, the original data vectors may be mapped to higher dimensional space to achieve linear separability again (Yiming and Xin, 1999). As the transformation into higher-dimensional space turns out to be quite computing intensive, different *kernels*, i.e. functions in lower-dimensional space that exhibit similar behaviour as the original functions in higher-dimensional space, have been proposed. Nonetheless, when the number of features is exceptionally large – as it is the case with text mining – using a linear *kernel* seems sufficient. Hsu, Chang and Lin (2007, p. 12) have shown that "the cross-validation accuracy using linear kernel is comparable to that using the [radial basis function] *rbf* kernel [and] hence, using a linear kernel is good enough". It is against this background that the authors use a linear *kernel*.

Evaluation of Classification Quality

In order to assess the applied learning algorithms' classification quality, the derived universal models need to be tested on an additional independent test dataset. As the underlying complete dataset turns out to be comparatively small, it seems unrealistic to conduct a one-time split into a training and a test dataset (e.g. 75:25, 50:50) without significantly reducing the features (information content) available for the learning task.

Therefore a k-fold (k=10) cross validation, which seems to be the preferred method of choice in situations with small datasets at hand (Witten and Frank, 2001), is conducted. Thereby, the complete dataset S is randomly split up into k subsets S_i. Having run trough k validations, each document within the corpus has been both part of the training and the test dataset. As highlighted by Lavrenko et al. (2000) ignoring the temporal ordering of the subsets should not present a problem.

The relevant performance measures (overall) accuracy, recall and precision are calculated for each round and are averaged at the end (*micro averaging*).

The overall accuracy is equivalent to the relative number of correctly classified examples, i.e. number of true positives and true negatives divided by the total number of documents (Robertson et al., 2007), and constitutes an accepted text mining related performance measure (Mittermayer, 2004). Given that two classes (e.g. "positive" and "negative") contain an equal number of documents, guessing would merely result in an accuracy of 50%. As the differences in the number of documents in each class turn out to be small in absolute terms, but comparatively large in relative terms, a *DefaultLearner* (Mierswa, Wurst, Klinkenberg, Scholz and Euler, 2006) algorithm is introduced and should thereon serve as a guessing-equivalent benchmark. The *DefaultLearner* creates a model based on a default value (e.g. mode of the true labels in case of classification) for all examples.

Depending on the final application of above models, i.e. may these primarily serve for buy and / or sell recommendations, the underlying performance needs to be scrutinized in more detail. For example, an algorithm classifying each document within an equally stratified training dataset as belonging to the category "positive" achieves the same overall accuracy of 50% as an algorithm classifying each document "negative" (Thomas, 2003). That is the reason why the concept of recall and precision is introduced.

$$precision = \frac{\text{Correct Classifier Assignments}}{\text{Total Classifier Assignments}} \tag{6}$$

$$recall = \frac{\text{Correct Classifier Assignments}}{\text{Total Correct Assignments}} \tag{7}$$

Learning Task "A"

As learning task A's major objective is to identify the news category with an expected abnormal price effect, the category under scrutiny is given by the class "financial statements". Having conducted 10-fold cross validation with *Classifier A,* it can be summarized that the results turn out to be rather promising.

Table 2 shows that the overall accuracy reaches a maximum of nearly 100%, or 94.38% on an (mirco) averaged basis. It follows that, compared to the *DefaultLearner*, the SVM performs better. Merely the recall ranges below 90%. Utilizing an additional (stop word) filter file and marginally altering the SVM inputs, however, resulted in a recall increase of ~ 5%. Overall, the two classes do not only seem to differ with regard to abnormal stock price reactions (Muntermann and Güttler, 2007), but also seem to be sufficiently different in nature to enable the learning algorithm to achieve above results. *Classifier A* may therefore either be used in isolation or as a pre-selector to global *Classifier B* and local *Classifier C* respectively (see Figure 2).

Experiment	Setup	*DefaultLearner*	Support Vector Machine		
Optimize Parameter:	Classification: News Category	Accuracy (micro)	Accuracy (micro)	Precision (micro)	Recall (micro)
None	Positive: fs Negative: nfs	61.88% +/- 1.87% (61.88%)	92.50% +/- 5.80% (92.50%)	97.50% +/- 5.00% (98.04%)	81.79% +/- 12.35% (81.97%)
C (SVM)	Positive: fs Negative: nfs	61.88% +/- 1.87% (61.88%)	93.12% +/- 7.76% (93.12%)	95.56% +/- 8.89% (96.30%)	85.00% +/- 14.34% (85.25%)
Filter file[1]	Positive: fs Negative: nfs	61.88% +/- 1.87% (61.88%)	93.75% +/- 4.84% (93.75%)	97.78% +/- 4.44% (98.11%)	85.13% +/- 9.81% (85.25%)
C (SVM) & Filter file	Positive: fs Negative: nfs	61.88% +/- 1.87% (61.88%)	94.38% +/- 5.38% (94.38%)	97.78% +/- 4.44% (98.11%)	86.67% +/- 11.30% (86.89%)

[1] The filter file serves as an additional stopword filter and contains names of towns, weekdays and months only.

Table 2. Learning Task "A" Results

Learning Task "B" & "C"

Taking both performance metrics *av* and *peak* into account, it can be observed that the global *Classifier B* (SVM) does make better predictions than the guessing-equivalent *DefaultLearner* (see Table 3). For instance, *Classifier B* is able to predict whether or not the returns will at least once during the 15 minute interval after news publication cross previously defined boundaries (2%, -2%), i.e. demonstrate abnormal price behaviour. The performance metric *peak,* however, may also be interpreted as weak form volatility measure.

Moreover, given performance metric *av*, the SVM again produces better results than the guessing-equivalent *DefaultLearner*; even though slightly worse than in the latter case. 0.09% is equal to average *av-15* of all companies in the dataset (see Table 3). In both cases, i.e. *peak* and *av*, local *Classifier C* performed alike, but not better.

The relative advantage of local *Classifier C* over global *Classifier B* can only be observed with the *sr* metric. Dividing the (sub) dataset into two classes, i.e. "positive" if *sr* is larger than zero during the interval, the local classifier surpasses both the *DefaultLearner* and the global classifier. Even though the local classifier exhibits a lower accuracy than the global classifier in the three-class case, especially the important recall measure is again much better. Hereby, it appears that negative news can be identified more precisely by our approach within the sub-group "financial statements".

Regarding whether or not this information may actually be of use, it shall be referred to those investors / decision makers that already hold a “long” position and would like to react to new (bad) news as fast as possible. Therefore above results offer first (weak) evidence that local classifiers may help to improve stock price prediction and consequently the proposed two-stage classification approach should turn out to be superior to simple one-stage global classification.

Nonetheless, it shall also be noted that the results need to be interpreted with caution as these are based on observations from a small dataset. Hypothesis $\boldsymbol{H_0}$ is therefore corroborated with proviso.

Classification		*DefaultLearner*	Support Vector Machine		
Metrics (Sample)	Classes	Accuracy (micro)	Accuracy (micro)	Precision[1] (micro)	Recall[1] (micro)
sr-15 (Whole)	Positive: > 0.00 Negative: <= 0.00	57.50% +/- 2.50% (57.50%)	55.00% +/- 6.12% (55.00%)	43.72% +/- 15.28% (44.44%)	23.63% +/- 8.62% (23.53%)
sr-15 (Sub-S.)	Positive: > 0.00 Negative: <= 0.00	47.62% +/- 5.22% (47.54%)	58.97% +/- 5.42% (59.02%)	60.87% +/- 8.87% (58.82%)	64.76% +/- 11.21% (64.52%)
sr-15 (Whole)	Positive: > 0.01 Negative: < -0.01 Neutral: rest	-	40.00% +/- 6.67% (40.00%)	31.37% +/- 8.00%	9.76% +/- 0.71%
sr-15 (Sub-S.)	Positive: > 0.01 Negative: < -0.01 Neutral: rest	-	34.62% +/- 8.66% (32.79%)	48.78% +/- 6.25%	54.45% +/- 13.73%
peak-15 (Whole)	Positive: > 0.02 Positive: < -0.02 Negative: < 0.02	55.62% +/- 1.87% (55.63%)	70.00% +/- 6.73% (70.00%)	68.79% +/- 5.89% (68.47%)	85.49% +/- 5.56% (85.39%)
av.-15 (Whole)	Positive: > 0.09% Negative: < 0.09%	52.50% +/- 3.06% (52.50%)	63.75% +/- 7.35% (63.75%)	62.48% +/- 7.23% (62.26%)	78.38% +/- 10.29% (78.37%)

[1] Recall and precision measures are given for the class of most interest: *sr* (“negative”), *peak* (“positive”), *av* (“positive”).

Table 3. Learning Task “B” & “C” Results

SUMMARY AND CONCLUSION

Building upon previous empirical research on abnormal intraday stock price reactions, the authors suggest two approaches on how to incorporate this knowledge within financial decision support systems.

Firstly, a “news category membership” classifier has been proposed. As the respective classification quality revealed to be very good and given above described importance of the category “financial statements” (Muntermann and Güttler, 2007), the resulting classifier already turns out to be of great use in isolation. Besides information filtering purposes, however, the classifier may also be applied within the proposed two-stage classification approach. It thereby serves as a pre-selector to either a global or a local classifier.

Secondly, a local classifier for the news category “financial statements” has been presented and compared to the respective global classifier. The results provide evidence that the local classifier makes better predictions than the global classifier. This finding highlights the importance of previous research within *supervised* learning. Knowing the examined classes’ characteristics very well seems to improve results. The two-stage classification approach, i.e. combining the “news category membership” classifier with local classifiers, should therefore improve results as well.

It is against this background that above findings need further investigation by means of a trading simulation. Moreover, a larger dataset needs to be gathered. Due to the fact that the focus on highly relevant (ad hoc) news provided promising results, the authors will extend research scope to other countries’ regulatory news as well.

REFERENCES

1. Brücher, H., Knolmayer, G. and Mittermayer, M. A. (2002) Document Classification Methods for Organizing Explicit Knowledge, in *Proceedings of the 3rd European Conference on Organizational Knowledge, Learning and Capabilities (ALBA)*, Athens.
2. Dhar, V. and Stein, R. (1997) Intelligent Decision Support Methods, Prentice Hall, Upper Saddle River.
3. Forman, G. (2003): An Extensive Empirical Study of Feature Selection Metrics for Text Classification, in *Journal of Machine Learning Research*, 3, 1289-1305.
4. Fung, G. P. C., Yu, J. X. and Lu, H. (2005) The Predicting Power of Textual Information on Financial Markets, in *IEEE Intelligent Informatics Bulleting*, 5, 1, 1-10.
5. Hsu, C. W., Chang, C. C. and Lin, C. J. (2007) A Practical Guide to Support Vector Classification, Working Paper, Natinal Taiwan University.
6. Hull, D. (1996) Stemming Algorithms: A Case Study for Detailed Evaluation, in *Journal of the American Society for Information Science*, 47, 1, 70-84.
7. Joachims, T. (1998) Text Categorization with Support Vector Machines: Learning with Many Relevant Features, in *Proceedings of the 10th European Conference on Machine Learning (ECML)*, Chemnitz, Germany, 137-142.
8. Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D. and Allan, J. (2000) Language Models for Financial News Recommendation, in *Proceedings of the 9th International Conference on Information and Knowledge Management*, November 6-11, McLean, VA, USA, 389-396.
9. Lewis, D. D. (1992) Representation and Learning in Information Retrieval, Dissertation University of Massachusetts, Amherst.
10. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. and Euler, T. (2006) YALE: Rapid Prototyping for Complex Data Mining Tasks, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*.
11. Mittermayer, M. (2004) Forecasting Intraday Stock Price Trends with Text Mining Techniques, in R. H. Sprague (Eds.) *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, Computer Society Press, Los Alamitos, CA, USA.
12. Muntermann, J. and Güttler, A. (2007) Intraday Stock Price Effects of Ad Hoc Disclosures: The German Case, in *Journal of International Financial Markets, Institutions and Money*, 17, 1, 1-24.
13. Patell, J. M. and Wolfson, M. A. (1984) The Intraday Speed of Adjustment of Stock Prices to Earnings and Dividend Announcements, in *Journal of Financial Economics*, 13, 2, 223-252.
14. Porter, M. F. (1980) An Algorithm for Suffix Stripping, in *Program*, 14, 3, 130-137.
15. Robertson, C., Geva, S. and Wolff, R. (2007) Predicting the Short-Term Market Reaction to Asset Specific News: Is Time Against Us?, in J. Z. Huang and Y. Ye (Eds.) *Proceedings Industry Track Workshop, 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2007)*, Nanjing, China, 1-13.
16. Schulz, A., Spiliopoulou, M. and Winkler, K. (2003) Kursrelevanzprognose von Ad-hoc-Meldungen: Text Mining wider die Informationsüberlastung im Mobile Banking, in W. Uhr, W. Esswein and E. Schoop (Eds.) *Wirtschaftsinformatik 2003: Medien, Märkte, Mobilität,* Band II, Physika, Heidelberg, Germany.
17. Thomas, J. D. (2003) News and Trading Rules, Dissertation Carnegie Mellon University, Pittsburgh.
18. Vapnik, V. (1995) The Nature of Statistical Learning Theory, Springer, New York.
19. Witten, I. H., Moffat, A. and Bell, T. C. (1999) Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd edition, San Francisco.
20. Witten, I. H. and Frank, E. (2001) Data Mining: Praktische Werkzeuge und Techniken für das maschinelle Lernen, Munich / Vienna.
21. Woodruff, C. S. and Senchak, A. (1988) Intradaily Price-Volume Adjustments of NYSE Stocks to Unexpected Earnings, in *Journal of Finance*, 43, 2, 467-491.
22. Wuthrich, B., Cho, V., Leung, S., Permunetilleke, D., Sankaran, K. and Zhang, J. (1998) Daily stock market forecast from textual web data, in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, October 11-14, San Diego, CA, USA, 2720-2725.

23. Wurst, M. (2007) The Word Vector Tool: User Guide, Operator Reference, Developer Tutorial, URL: “http://wvtool.sf.net”, [Access date: December 2007].

24. Yiming, Y. and Xin, L. (1999) A re-examination of text categorization methods, in *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, 42-49.