Classifying Press Releases and Company Relationships Based on Stock Performance[*]

Mike Mintz
Stanford University
mintz@stanford.edu

Ruka Sakurai
Stanford University
ruka.sakurai@gmail.com

Nick Briggs
Stanford University
njbriggs@stanford.edu

Abstract

We classify press releases as “good” or ‘bad’ news for 3 companies based on whether the stock increases n minutes after publication. We tried different classifiers (Multinomial Naive Bayes, Regularized SVM, and Nearest Neighbors) and various feature representations (such as the TF-IDF of the words in the document). We do a few percent better than majority baseline with our best setup: nearest neighbor classifier with a cosine similarity metric, binary word-in-doc features, and $n = 15$ minutes. Stemming words to base forms helped significantly. Using the clustering to predict the stock price of related companies did not work. Overall a lack of sufficient press release data was the limiting factor of our research. Various suggestions for improvement are discussed in the conclusion.

1 Introduction

Press releases are usually the first time news about companies is made available to the public. We therefore hypothesized that the contents of the press releases are a majority indicator of the short term value of a company’s stock. A machine learning approach would be able to analyze these press releases and make predictions about the stock price much faster than a human analyst could. Such a tool could aid a trader in making quicker decisions based on press release information, and also help classify press releases as “good” or “bad” news for a particular company.

We compiled a large corpus of press releases for publicly traded companies, as well as a corpus of stock price changes for these companies, with high time precision. We created a classifier for these articles, and trained it using the short-term percent change in stock price for the company. Then, given a press release when it is announced, our classifier attempts to predict whether the stock price of its company will increase or decrease in the short term.

1.1 Prior Work

Previous work in this area was performed by Mittermayer [4], who designed a system to analyze press releases in real time and make stock transactions decisions based on them. He used an SVM and reported that the SVM had trouble marking press releases as “good news” or “bad news”.

2 Data Collection

2.1 Stock Data

Through the Graduate School of Business Library, we collected stock data from the New York Stock Exchange Trade and Quote Database (NYSE TAQ) provided by University of Chicago’s Center for Research in Securities Prices (CRSP). We focused on intraday data about all companies in the NYSE. For the intraday data we retrieved the price, volume and time (to the second) of all trades that occurred. Typically there are multiple transactions that occur within a minute. This provides us with stock values that are highly precise with respect to time. This data is also used for clustering companies with similar market fluctuations.

[*] Thank you Dan Ramage for the great advice for text classification!

A month's worth of data for all companies in the NYSE comprises more than 10 gigabytes of information. Therefore the challenge is to store this data in an efficient way without losing the precision that is needed in our analysis. Since press release times are recorded with precision to the nearest minute, we store the stock value at each minute. The stock value at a specific minute is calculated by the weighted average of the trades that occurred in that minute, where the weights are the volumes of the transactions. The value of the stock at times without data from the NYSE TAQ are computed by taking the value from the nearest minute that has price information.

2.2 Article Data

We retrieved press releases and news articles from the Factiva system through the Graduate School of Business Library. We focused on press releases from 2006 and 2007, since there was a lot less data available for other years. To simplify the problem, we limited ourselves to classifying press releases from three large companies: Boeing, McDonald's, and Verizon.[1]

The press releases were available as XML files, and contained information about the title, date, paragraph structure, and other metadata that Factiva used for indexing. We simply stored the date and calculated a set of all words contained in the article. All letters were converted to lowercase, punctuation was removed, stop words were dropped, and we did some generalization by replacing specific numbers with generic "number" tokens.

Articles are kept only if they have date and time information fully set. Some articles only have a published date, which makes it impossible to associate them with stock price changes during the day. At our milestone, we had a lot of noisy articles in our database that were not actually press releases, since Factiva's "press release" classification was not very accurate. By identifying the most common distributors of true press releases in our corpus, we were able to remove this noise.

We also test the publication date against the stock data to make sure there were trades going on around that time. Articles that do not have any trades between its publish time and 15 minutes later are discarded. This brings the number of articles down from 3583 articles with full time information to 2690.

Over our entire corpus of articles, our vocabulary size is about 27,000 (after lower-casing words and removing stop words and numbers). We incorporated a word stemmer [5] into our project to convert every word to its base form. For example, it converts both "running" and "run" to "run", and reduces our vocabulary size to about 19,000 (30% fewer features). As described in our results, this helps our accuracy significantly.

We wanted to identify bigrams (and possibly higher-order phrases), since phrases like "high profit" are only recorded as "high" and "profit", each of which on its on is not particularly correlated with good or bad news. We tried adding all seen bigrams as features, but because of the large number of unique bigrams used in our entire corpus, we had a data explosion and could not store the feature vectors for even one company in memory.

3 Classification

3.1 Implementation

A press release was categorized as "good news" if it preceded a rise in stock price over the next n minutes, and "bad news" otherwise. We associated each press release with stock trade data in the appropriate window. We trained on 80% of our data for each company (selected randomly using a consistent random seed) and tested on the remaining 20%. We implemented and trained three classification algorithms: Multinomial Naive Bayes (NB), Support Vector Machine (SVM), and Nearest Neighbor (NN).

Our implementation of NB was based on [1]. Since we only had 2 categories, we did not implement Complement Naive Bayes as described in the paper. Instead, we implemented category weight normalization, document length normalization, text frequency adjustment (using the power law distribution $\log(1 + f_i)$, where f_i is the number of occurrences of a term in the document), and inverse document frequency.

[1] Since we train a separate classifier on each company, it would not improve our performance to gather data from more companies, but doing more than one allows us to do better error analysis.

To implement the regularized SVM, we adopted the LIBSVM library [2].

NN was suggested by [3]. At first, we tried to calculate distances by using the Euclidean norm. Later testing showed that max cosine similarity $\frac{u \cdot v}{||u||_2||v||_2}$ gave the best results.

In addition to these three classifiers, we also implemented a voting classifier that trained these 3 classifiers, and used a majority vote to make a prediction (weighted by the confidence of each classifier that supported probabilistic predictions).

3.2 Features

We started out by using tokens from press releases as-is. One of the first things we added to increase accuracy was a stemmer [5], reducing the feature set size by removing different forms of the same word. As we experimented, we began to take into account document length, term frequency in a document, and the inverse document frequency of terms (it is assumed that especially important individual terms appear in few documents, hence *inverse* document frequency). For our final round of tests, we had four configurations for NN and SVM: existence of a term in a document, the count of a term in a document, the count of a term divided by the document's length (normalized word count), and TF-IDF (normalized word count times a term that penalizes words that appear in many documents). For NB, the features mentioned in [1] were always used.

4 Classification Results

A comparison of the classification accuracy of various algorithms and feature types are shown in Figure 1. The vertical axis shows how much more accurate the results were compared to a majority baseline classifier. The majority baseline classifier classifies all examples as the most frequent class in the training set. In this case since the stock market increased on average in our corpus, the majority baseline classified press releases as positive. The majority baseline classified with a 51-53% accuracy.

Among the various algorithms, NN performed the best, followed by SVM. Computing the feature values as a binary Word-In-Doc out performed the other methods. Normalized word count and TF-IDF performed below majority baseline. It is especially surprising that the normalized word count performed significantly worse than the unnormalized word count. It's possible that our classifiers were taking advantage of the document length being an important feature, and by normalizing the word counts, we removed this information from our features.

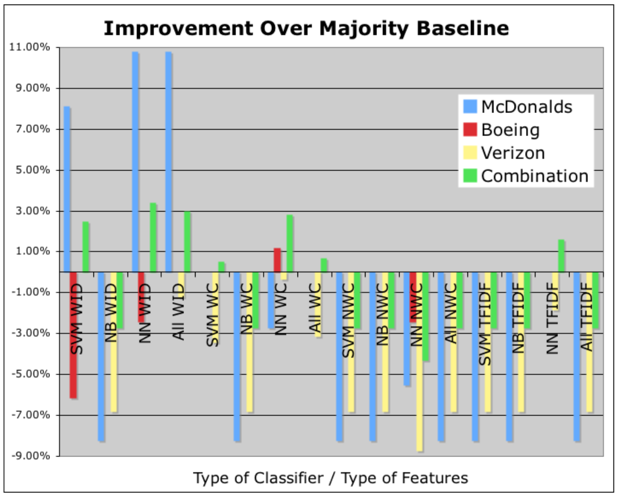

Figure 1: The performance of classification with various algorithms and feature types. Algorithms:(SVM-Support Vector Machine, NB- Multinomial Naive Bayes, NN- Nearest Neighbors All-Combination of three algorithms) Feature Types:(WID- Word In Document, WC-Word Count, NWC- Normalized Word Count, TFIDF-Term Frequency Inverse Document Frequency)

The classifiers worked best for classifying the press releases of McDonald's. The nearest neighbor classifier with features represented as a binary word in document classified McDonald's press releases 11% better than the majority baseline classifier.

Press releases are written by each company itself, so it is reasonable that our algorithms perform differently for different companies. The press releases on some companies may have a very neutral tone at all times, using very similar vocabulary. On the other hand the press releases of other companies may vary its vocabulary significantly between publications. The positive correlated features of McDonald's (according to Naive Bayes weights) were mostly related to its service such as 'variety', 'foodservice', and 'customers.' On the other hand, the negative features of McDonald's seem to be related to finance such as ‘share’, ‘outlook’, and ‘report.’ This might suggest that press releases announcing news related to its services correlates with an increase in stock price, whereas press releases announcing financial information correlates with a decrease in stock price.

Positive	Negative
variety	now
over	common
visit	shares
llc	full
ingredients	outlook
foodservice	you
inc	open
restaurants	related
customers	report
through	stock

Figure 2: Most important features for McDonald’s

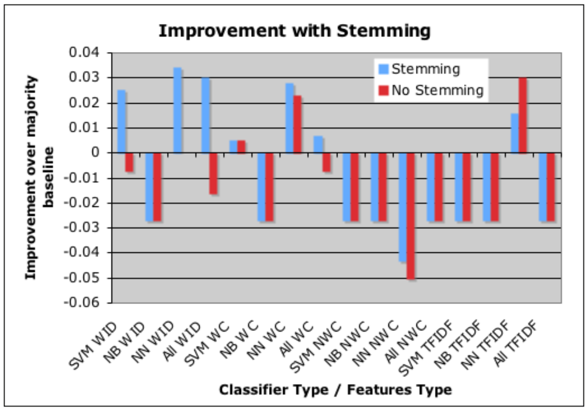

Figure 3: The effect of stemming on accuracy.

Figure 3 shows how stemming improved our classification accuracy. For each classification method, the results when stemming is used outperform the results when stemming is not used in all but one case. Stemming reduces the feature size. The improvement in performance may be due to reduced overfitting by decreasing the feature size.

The algorithm depends on the time it takes for the stock market to respond to a press release. Tests were performed with various assumptions about the stock market response time. Some of the results of these tests are shown in Figure 4. The graph shows the performance of the nearest neighbor classifier (using word-in-doc) as a function of various response times. The best performance was observed when the stock market response time was assumed to be 15 minutes. Without more data and test results, the difference in accuracy may not be significant enough to make a confident conclusion about the response time of the stock market to a press release.

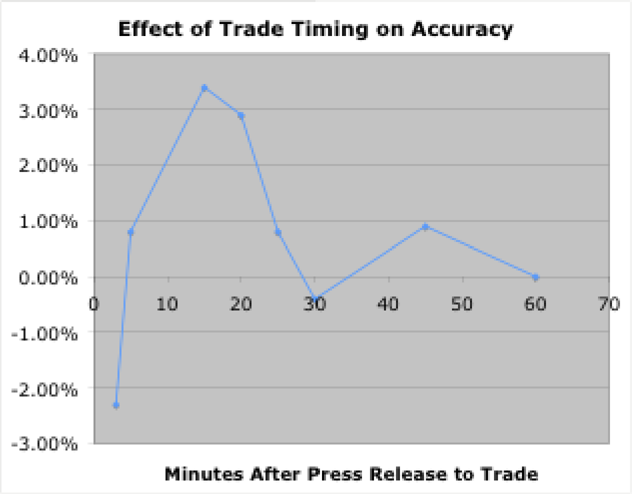

Figure 4: The effect of trade timing on accuracy. Trade timing is how long the algorithms waits after the press release publication time to compute the change in stock price.

5 Clustering

We implemented a clustering algorithm to find stocks that perform similarly. We obtained one month of stock trades for every company available in TAQ. We discretized the average trade price by the hour, and for every hour from the beginning to the end of the month, we calculated the percent change in price for every stock from the previous hour. Thus, for every stock, we had feature vectors with about 700 features representing the direction of stock movement.

At first we clustered the stocks using K-Means, but no matter how high k was, there were always some very large clusters. We simplified the algorithm by just finding the k closest stocks for each stock (using the same Euclidean distance metric). As validation for the success of using percent changes every hour, we noticed that the closest company to Boeing was Rockwell Collins, an independent branch of a company that Boeing bought several years ago. Also, we found that for oil companies like Exxon and BP, other oil companies were in its cluster, which makes sense since their stock prices are all dependent on a single variable for the price of oil. However, most of highly related companies were big investment companies that we had never heard of, which are probably correlated with the companies because they invest in them.

Specifically, we looked at the 2-3 closest stocks to McDonald's, Verizon, and Boeing. For each related company, we trained a new classifier for its stock price, based on the press releases of the original company (e.g., McDonald's). However, on our best classifier setup, the accuracy of the related companies was in general significantly worse than the accuracy of the original companies, and in 4 of the 5 related companies, was worse than majority baseline. This suggests that short term stock changes are not correlated very well with related companies, which is an unfortunate result, but it also tells us that the features we got from the press releases are actually meaningful to the company they were trained for – at least, meaningful enough that the classifier performs worse on data from other companies.

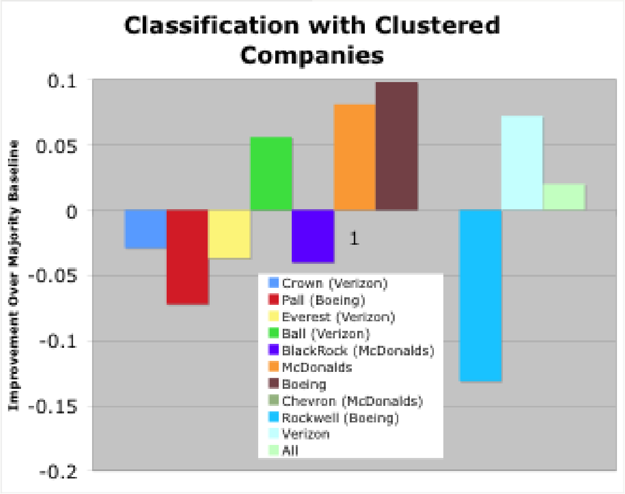

Figure 5: The performance of related companies vs. original companies with NN WID. Parenthetical companies are the original companies they are clustered with.

6 Conclusion

Although we had positive results after fine-tuning our classifier setup, we believe that a lot of our negative results are due to most of the press releases actually being uncorrelated with changes in the stock market. Changes in the stock market only happen when investors get new information that affects their judgment about the profitability of the company, and many articles might not actually provide information to this effect. Upon further analysis of our press releases, only 13 of the 2690 press releases that happened during trading hours saw a 1% or higher stock price increase. Lowering our standards to change in either direction by at least 0.1%, we found that only half of the articles have this. We tried considering only examples where the stock price rose above a threshold percent positive, and the rest negative, but this only lowered our accuracy because of very few positive examples. Since we depleted our source of press releases for 2006 and 2007, it may not be possible to get more data. But what might help is having our system analyze more volatile stocks, since more exciting news tends to be announced which can surprise investors. We wanted to find NASDAQ data but the best database we found at the library was the NYSE TAQ.

As possibilities for further research, we could decrease the number of features and add more variety to the type of features. We could decrease our features by ignoring words that appear with approximately equal distribution in positive and negative examples, and more advanced word clustering (in addition to stemming, we could use WordNet to collapse synonymous words to the same feature). We could increase the variety of the types features by adding other metadata about press releases and stocks, such as the change in stock price before the press release was published and the number of words in the press release, as well as bigrams (without those that appear rarely or in equal distribution among positive and negative examples). Finally, we could try reducing noise by removing the overall change in the stock market from the change in price, so that external effects like interest rate cuts have less effect on our data. We could use a clustering algorithm to divide the companies by industry so that we could train companies differently based on their industry.

7 References

1. Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, David R. Karger, “Tackling the Poor Assumptions of Naive Bayes Text Classifiers” in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington, D.C., 2003. Available HTTP:
http://groups.csail.mit.edu/haystack/papers/rennie.icml03.pdf

2. Chih-Chung Chang, Chih-Jen Lin, *LIBSVM - A Library for Support Vector Machines*, Available HTTP:
http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

3. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schtze, *Introduction to Information Retrieval*, Cambridge University Press. 2008. Available HTTP:
http://www-csli.stanford.edu/∼hinrich/information-retrieval-book.html

4. M.A. Mittermayer, “Forecasting Intraday Stock Price Trends with Text Mining Techniques” in *Proceedings of the Hawai'i International Conference on System Sciences*, January 5-8, 2004, Big Island, Hawaii. Available HTTP:
http://www.ie.iwi.unibe.ch/staff/mittermayer/resource/Hawaii.pdf

5. Martin Porter, *Snowball*, Available HTTP:
http://snowball.tartarus.org/