A Sentiment Detection Engine for Internet Stock Message Boards

Christopher C. Chua [†‡] **Maria Milosavljevic** [‡] **James R. Curran** [‡*]

School of Computer Science and Engineering[†]
University of New South Wales
NSW 2052, Australia

Capital Markets CRC Ltd [‡]
55 Harrington Street
NSW 2000, Australia

School of Information Technologies [*]
University of Sydney
NSW 2006, Australia

{cchua,maria}@cmcrc.com james@it.usyd.edu.au

Abstract

Financial investors trade on the basis of information, and in particular, on the likelihood that a piece of information will impact the market. The ability to predict this within milliseconds of the information being released would be useful in applications such as algorithmic trading. We present a solution for classifying investor sentiment on internet stock message boards. Our solution develops upon prior work and examines several approaches for selecting features in a messy and sparse data set. Using a variation of the Bayes classifier with feature selection methods allows us to produce a system with better accuracy, execution performance and precision than using conventional Naïve Bayes and SVM classifiers. Evaluation against author-selected sentiment labels results in an accuracy of 78.72% compared to a human annotation and conventional Naïve Bayes accuracy of 57% and 65.63% respectively.

1 Introduction

In this paper, we present a sentiment prediction engine for classifying investor sentiment, i.e. signals to buy, sell or hold stock positions, based on messages posted on internet stock forums. Our sentiment annotated corpus comes from HotCopper[1], the most popular investment forum for the Australian market, where posts include author self-reported sentiment labels. This unique characteristic of this data set present us with an opportunity to extend research in sentiment classification. Our automated sentiment detection engine implementation uses variations classifiers, particularly the Bernoulli Naïve Bayes and the Complement Naïve Bayes (CNB) models, coupled with feature selection techniques using InfoGain, phrase polarity counts and price alerts. Our methods achieve 78.72% accuracy for CNB and 78.45% for Bernoulli. These figures are higher than the 57% accuracy from human annotators and 65.63% in the baseline. It also outperforms results from Das and Chen (2007) on a different dataset.

[1] http://www.hotcopper.com.au

2 Problem Domain

Our results contribute towards the development of a real-time solution which monitors financial information in order to provide useful advice to support market surveillance analysts' task of explaining alerts surrounding price movements in stocks. For example, when the overall sentiment towards a particular stock is positive, it may well explain the observed increase in its uptake. Many forums do not provide an ability for authors to explicitly report sentiment, thus we hope to eventually apply this model to other forums. The "Buy", "Hold" and "Sell" tags are analogous to positive, neutral and negative sentiments respectively. HotCopper also includes a finer-grained labeling system for "short term" and "long term" sentiments. However such distinctions are beyond our current scope because a finer granularity in the recommendation strength given limited contextual information is often established through an in-depth knowledge of underlying financial fundamentals or information related to a particular stock not reflected within a short message text.

Classifying investor sentiment based on web forum messages is a challenging problem in the text classification domain. The dataset is not only sparse, but varies in the overall quality of its labels and descriptive content. For instance, the sentiment labels are likely to vary in a thread from one post to another, which indicates disagreement. Previous work on sentiment classification is based around relatively well-formed texts (Durant and Smith, 2006; Pang et al., 2002). As demonstrated in Milosavljevic et al. (2007), information extraction techniques such as sentence boundary detection and part-of-speech tagging work relatively well on structured texts but perform less well on messy and sparse data sets such as forum posts and interview transcripts. Hence, we require the use of techniques beyond conventional approaches. Furthermore, the constraints of a real-time classification system presents additional challenges.

3 Background

As the literature directly related to this domain is limited, we draw from related areas of sentiment classification research where a research efforts have been concentrated around sentiment or opinion analysis for political blogs (Durant and Smith, 2006) and product reviews (Yi et al., 2003). The methods developed in those prior work are relevant to our application.

Sentiment analysis on web forums specifically within the financial domain has also been investigated by Das and Chen (2007). Their focus, like ours, is on capturing the emotive aspect of the text rather than the factual content. In their research, the Naïve Bayes (NB) classifier is found to yield the best results, and a voting mechanism is used in conjunction with additional classifiers such as SVM to improve accuracy. However, the classification accuracy achieved at 62% using a simple majority vote of multiple classifiers with a small sample and the low inter-annotator agreement demonstrate the difficulty in classifying such datasets. Antweiler and Frank (2004)'s research findings found that online forum discussions between investors are not equivalent to market noise, and instead contain financially-relevant informational content. As a result, effective sentiment detection can predict market volume and volatility across stocks, thus highlighting the need for placing such web discussions under the investigative eyes of surveillance analysts. Both Das and Chen (2007) and Antweiler and Frank (2004) use data from Yahoo Finance and Raging Bull based in the US, covering only a subset of stocks, with classification performed per stock rather than in aggregate.

The prior literature demonstrates that the sentiment analysis task can be performed using a variety of classification methods, chief among them the NB model (Das and Chen, 2007; Antweiler and Frank, 2004). Similar to Das and Chen (2007) and Antweiler and Frank (2004), we find that a typical SVM classifier performs no better than the alternatives we attempted, while suffering from a higher degree of complexity affecting execution performance. Moreover, prior solutions presented do not offer a comprehensive sentiment analyser to predict sentiment off financial forums in real-time for market surveillance or technical trading. We extend the concepts presented in prior research by incorporating additional contextual information in our training tasks, developing more advanced feature selection as well as adopting variations of the models used in related research.

Statistic	Buy	Sell	Hold
Total	6379	469	1459
Monthly Average	1063.17	78.17	243.17
Monthly Std Dev	283.65	28.22	50.95

Table 1: HotCopper Post Statistics

4 Data

In our analysis, we use the first six months of 2004 HotCopper ASX stock-based discussions. There are 8,307 labeled posts across 469 stocks, with an average of 28 words per post and a total of 23,670 distinct words in the dataset. Each message is organised by thread, with a single thread consisting of multiple posts on the same topic for a stock. We consider both long term and short term variations of a sentiment to be equivalent. "Buy" recommendations outnumber "Sell" and "Hold" 13.6 and 4.4 times respectively. Within first 18 months of our analysis, the average monthly posts increased from over 1,400 to a peak of over 3,700 posts by August 2005, indicating growing forum participation. Discussions on HotCopper mainly surrounds speculative stocks, particularly those in minerals exploration and energy. In

fact, some of the biggest stocks by market capitalisation on the ASX such as the Commonwealth Bank (CBA) and Woolworths (WOW) generate little to no active discussions on the forums, highlighting the focus on small and speculative stocks.

4.1 Data Preprocessing

We perform a series of preprocessing steps for each post to obtain a canonical representation, firstly by removing stop words from the training set in the NLTK stop list (Bird et al., 2009). Words and alphanumerics of non-informative value, e.g. "aaaaaa" or "aaaaah", are filtered out, the remaining stemmed using the Porter algorithm (Porter, 2009) with spell-correction applied using the first suggestion from the PyEnchant package (Kelly, 2009).

We observed many ambiguous instances which introduce noise to the training model. In order to control for this, a thread volatility measure is introduced for the message where we assign an index value representing the level of disagreement between subsequent replies in the thread. The thread volatility is measured as the average sum of the differences between the discretised values of the sentiment classes. We assign buy and sell to have the furthest distance, thus the discretised set S contains $\{$buy=1,hold=2,sell=3$\}$. Threads which transitions from buy to sell result in a higher volatility figure than threads which transitions from buy to hold. This allows for the posts within a thread with lower volatility to emerge as a superior sample. The thread volatility measure for a discretised sentiment s_i in thread t with N_t posts, is defined as follow:

$$\sigma_t = \frac{1}{N_t} \sum_{i=1}^{N_t - 1} |s_{i+1} - s_i|$$

We select threads with low volatility (< 0.5) for our training base in order to reduce the level of disagreement in the training set. This filtering step reduces our effective sample size to 7,584 and enhances the quality of the training sample.

5 Classification

Our first experiment consisted of a baseline NB classifier (McCallum, 1998). The NB classifier follows Bayesian probability theory in selecting the maximum likelihood of an outcome given its prior probabilities. We are interested in the most probable class (MAP), given a message instance d with n features f and set of sentiment classes S:

$$MAP = \arg\max_{s \in S} P(s) \prod_{i=1}^{n} P(f_i|s)$$

A simplifying assumption is to treat the presence of individual features in the message d containing n words as positionally-independent of other words in the document. Although weakly-formed, this is found to perform well due to its zero-one loss property (Domingos and Pazzani, 1997). Laplace's add-one smoothing method is used to account for zero probabilities.

Following this, we tested an adapted version of the NB classifier to improve our classification accuracy, by incorporating the Term Frequency Inverse Document Frequency (TF-IDF) transformation (Rennie et al., 2003), which allows us to weigh terms that provide a greater distinction to a particular post more heavily than ones which appear with regular frequency across all posts and are poor features to rely on for classification.

$$TF - IDF_{f_i} = \ln\left(\sum f_i + 1\right) \ln\left(\frac{\sum_j d_j}{\sum_j d_{j,s \in S}}\right)$$

Another issue that we have to contend with is the uneven class distribution in the dataset, which is a common issue in text categorisation. Undersampling or oversampling methods results in an inaccurate distribution of underlying data, hence to overcome this limitation, we apply the approach used by Rennie et al. (2003) to tackle this skewness. The CNB classifier improves upon the weakness of the typical NB classifier by estimating parameters from data in all sentiment classes except the one which we are evaluating for. For a given message j with n features f, the CNB classifies documents according to the following rule:

$$l(f) = \arg\min_{s \in S} \sum_{i=1}^{n} f_i w_{s_i}$$

f_i is the count of feature i in the post and w_{s_i} is the complement weight parameter which is the TF-IDF transformed complement of the likelihood estimates (see Rennie et al. (2003)).

Finally, we also tested the classifier performance with the Bernoulli model of Naïve Bayes (McCallum, 1998), which replaces feature frequency counts with Boolean values. The use of the CNB classifier

Classifier	NB Baseline	CNB	CNB IG	NB Binarised	NB Binarised IG
# of Features	7,200	7,200	1205 (Rank 50)	7,200	1205 (Rank 50)
Accuracy	65.63%	74.41%	**78.72%**	75.75%	78.45%
Precision	68.50%	74.80%	**76.70%**	70.30%	73.40%
Recall	65.60%	74.40%	78.70%	75.80%	**78.50%**
F-score	66.90%	74.50%	**77.50%**	72.00%	72.00%

Table 2: Results Summary

and Bernoulli variant yields a statistically significant improvement in the classification accuracy, which is consistent with the findings of Pang et al. (2002) in the sentiment analysis domain.

6 Feature Selection

The features are first ranked by order of frequency. An optimal set of features is selected by testing feature increments up to a maximum of 10,000 features; approximately 40% of the base. We then tested the information gain (InfoGain) algorithm (Yang and Pedersen, 1997), which is useful in filtering out the vast number of features to a manageable subset. Among the additional features we incorporate is the count of positive and negative bigrams and trigrams (including negations) of the form "ADJ financial_term" where financial terms are common phrases encountered within the sample such as "EPS", "dividends" and "profit" representing domain-specific knowledge. Another domain-specific feature we incorporate is the count of stock price alerts in the 3 days preceding the start of a thread. A price risefall alert is triggered when the stock price risesdrops beyond 4 standard deviations from its historical price change levels.

7 Results and Evaluation

In any machine learning task, it is crucial to verify our results against human agreement levels. We took a random sample of 100 opening posts (to avoid out of context replies) and published an annotation task using Amazon's Mechanical Turk (MTurk) (Amazon, 2009) to obtain classifications from three paid annotators who passed a test. The disappointingly low annotator accuracy of 57% and Kappa agreement of 50% demonstrates the challenging nature of this task, even for humans.

We perform each experiment using 10-fold cross-validation and compare the performance based on accuracy in conjunction with F-scores. Table 2 summarises our main findings in terms of sentiment classification quality. At 7,200 features, the best performance is seen in the CNB and Bernoulli classifiers. In both schemes, InfoGain attribute selection improved F-scores by 10.60% and 5.10% respectively with 1,205 features compared to the baseline. The overall accuracy of both classifiers, at 78.72% and 78.45% are significantly above those attained in the baseline.

Our results reveal two classification strategies in our implementation, i.e. using either the CNB or the Bernoulli NB model. We also find that feature selection techniques and filtering noisy instances with the volatility measure, increase overall performance to a level higher than that of the baseline. Positive and negative phrase counts do not yield significant improvements in performance, which could be explained by a change in sentiment tone as evidenced in Pang et al. (2002). For example, a post may be labeled "Sell" but contain positive messages unrelated to the subject. This may be improved by using entity recognition to disambiguate context. Further extension that we hope to incorporate into the classification model is the addition of financial information reported in the media to help augment information not reflected in the message board post.

8 Conclusion

We introduce a sentiment prediction engine that allows for the real-time classification of sentiment on internet stock message boards. Through the application of alternative models and additional feature selection schemes, we are able to achieve classification F-score of up to 77.50%. We believe that more advanced natural language processing techniques, particularly deeper contextual analysis using external sources of financial data as well as improving the handling of imbalanced classes, will provide fruitful grounds for future research.

References

Amazon. 2009. Amazon Mechanical Turk. http://aws.amazon.com/mturk.

W. Antweiler and M.Z. Frank. 2004. Is All that Talk just Noise? The Information Content of Internet Stock Message Boards. *Journal of Finance*, pages 1259–1294.

S. Bird, E. Loper, and E. Klein. 2009. Natural Language Toolkit. http://www.nltk.org.

S.R. Das and M.Y. Chen. 2007. Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. *Journal of Management Science*, 53(9):1375–1388.

P. Domingos and M. Pazzani. 1997. Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. *Machine Learning*, 29(2–3):103–130.

K.T. Durant and M.D. Smith. 2006. Mining Sentiment Classification from Political Web Logs. In *Proceedings of Workshop on Web Mining and Web Usage Analysis of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (WebKDD-2006), Philadelphia, PA*.

R. Kelly. 2009. PyEnchant Spellchecker. http://www.rfk.id.au/software/pyenchant/.

A. McCallum. 1998. A Comparison of Event Models for Naïve Bayes Text Classification. In *AAAI Workshop on Learning for Text Categorization*, pages 41–48.

M. Milosavljevic, C. Grover, and L. Corti. 2007. Smart Qualitative Data (SQUAD): Information Extraction in a Large Document Archive. In *Proceedings of the 8th RIAO Conference*.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs Up?: Sentiment Classification using Machine Learning Techniques. In *Proceedings of the ACL conference on Empirical Methods in Natural Language Processing*, volume 10, pages 79–86. Association for Computational Linguistics.

M. Porter. 2009. The Porter Stemming Algorithm. http://tartarus.org/~martin/PorterStemmer/.

J.D. Rennie, L. Shih, J. Teevan, and D. Karger. 2003. Tackling the Poor Assumptions of Naïve Bayes Text Classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 616–623.

Y. Yang and J.O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420.

J. Yi, T. Nasukawa, R. Bunescu, W. Niblack, I.B.M.A.R. Center, and CA San Jose. 2003. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. In *Third IEEE International Conference on Data Mining*, pages 427–434.