Stock Market Prediction Based on Public Attentions: a Social Web Mining Approach

Ailun Yi

Supervised by Miles Osborne

Master of Science

School of Informatics

University of Edinburgh

2009

Abstract

In this thesis, we put the problem of stock market prediction in the new context, namely, the social media. Social media is a new form of content on the Web. One of its major characteristics is the timely provision of new content and quick interaction among users. Such intensive publishing and interaction can be viewed as a measure of users attention towards a large range of topics including stock-related information. As we hold the assumption that retrieving such information timely and analysing the level of attentions in a large scale can reveal interesting relationships to the stock prices, we evaluate various methods in representing such information including simple frequency counting, loose n-gram models and noun phrase expansion. Our results showed that although the simplest counting method failed to correlate directly with stock prices, models built on more complex features that bring cross-related concepts are shown to be effective in prediction.

Acknowledgements

I would like to thank my supervisor, Miles Osborne, for his enormous help and advice throughout the project. Many thanks go to my family standing behind me in every possible situation. Special thanks also go to Dan Harvey and Chen Wang for their great help in shaping this thesis.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Ailun Yi*
Supervised by Miles Osborne)

Table of Content

List of Figures

Chapter 1

Introduction

Stock market prediction is an area that has been gaining long-time interests across the fields of finance and many research communities. According to the efficient-market and random walk hypotheses (Fama, 1965; Malkiel, 1973), the denial of predictability of the financial markets has stood for a long time while many techniques are invented in order to refute its validity. Technical analysis believes that historical prices and other indicators can reveal correlations and patterns of stock price movements and leads to the predictability of future prices. In recent years, more advanced computational techniques such as artificial neural networks, genetic algorithms and support vector machines make the analysis and learning of such patterns more popular. From another realm, these machine learning techniques have been seen successfully application on text-based tasks such as information retrieval, text categorisation and clustering etc. By holding the belief that stock prices are highly correlated and sensitive to unseen news and events (Mackinlay, 1997) which bring new information for the market to absorb, approaches utilising learning scheme to find out such correlations and impact from past in order to predict future reactions on news are devised (Fung, Yu and Lam, 2002; Lavrenko et al. 2000).

In this thesis, the stock market prediction problem is addressed based on textual approaches. However, we attempt not to continue prediction based on news data as modelling different reactions and weighting the impact on the news and upcoming events are shown to be hard to implement (Yoo, Kim and Jan, 2005). Instead, giving rise of a new form to interact with the Web, social media is transforming the way information are provided and propagated over traditional media. We put the assumption based on our observation that information on social media, especially popular topics and news, are propagated quickly and can attract vast amount of attentions. If such information has an impact on stock prices, the vast attention may encode

several possibilities and correlations with the prices which we can possibly retrieve and model.

Another novel attempt in this thesis is the task nature shifted from previous works. Most previous works on news based system treat the prediction task as a classification problem, in which a given news article or event data is classified into trend classes indicating their impact on stocks. In this thesis, we firstly set our task to predict the real-valued price variable for a given stock through regression. In addition, as financial markets could be efficient in absorbing information from many channels, predicting upon standalone news articles or headlines could rarely capture the big picture. Alternatively, we view the task from an information retrieval perspective. We firstly assume our collection of data is large to contain enough information and then aim at retrieving the set of data for a stock query on a specific day, in which the features in the retrieved dataset have strong correlations with prices of the querying stock. Modelling on such sample set can give more accurate prediction to the future prices.

In realistic, there is no data collection that contains all necessary stock related information for retrieval purpose. Therefore, in the experiments of this thesis, we focus only on the social media data based on our initial assumption that the vast amount of attentions assembled timely on this new platform with their unique characteristics can provide a big picture of the events and other factors that affect stock prices.

To best of our knowledge, there is no previous work directly implementing similar goals with the new assumptions and data source. Therefore, our experimental results can be primitive at this stage but we aim at evaluating a wide range of settings in the current context where the future research can benefit from. We mainly explore the area of feature filtering and selection methods from the current dataset in order to understand the effect of different types of features with respect to their correlations to the stock prices and the performance of the learnt models. Our results show that the experiments yield a number of interesting findings with both positive and negative prediction performance. The models built upon our current settings can perform well when there are sufficient attentions and mentions regarding to the target stock in a timely manner. However, in the current status of the social media it is not possible to retrieve information that can help predict in any stock. Therefore, further justification to the assumptions can be drawn for future works.

The rest of the thesis is structured as follow. Chapter 2 firstly presents a background review on related theories and knowledge. A more detailed introduction and discussion on the social media and its specific forms are given in Chapter 3. The conceptual design of experiments is outlined in Chapter 4, followed by an evaluation on the experiment results in Chapter 5. Chapter 6 summarises the findings and sketches several directions for future research.

Chapter 2

Background

In this chapter the background theories and knowledge are reviewed in order to provide the necessary support for exploration of the subject area. We firstly introduce some key ideas in finance with respect to the market and information efficiency. Then turn the direction to the main tools for our prediction task, namely the support vector machine algorithm and feature selection methods.

2.1 Information and Efficient Market Hypothesis

In financial markets, reliable information is the key. The primary economic function of a financial market is to channel funds between firms or individuals for exchanging and sharing productive investment opportunities. The *efficient-market hypothesis* (EMH) (Fama, 1965) asserts that the market is "informationally efficient", i.e. stock prices always reflect all known information and the future's price movement is only in response to unseen news or events. Further derivation of the hypothesis breaks into three forms: weak, semi strong and strong.

The weak form of EMH emphasis that only "past" information is embedded in the current price and thus they are fruitless in term of predicting the future. Under this form of EMH, *technical analysis*, the study of historical stock prices in an attempt to predict future ones, will not be able to generate excess returns in the long run but *fundamental analysis*, the analysis of financial information such as company profit, asset values, etc. may still produce risk-adjusted rates of returns. The semi-strong form of EMH extends the weak form to imply that the market is efficient enough to reflect in current prices all public available information, such as the company's fundamentals, earning announcement, and all pubic news, etc. Public information is assumed to be available to everyone and fundamental analysis is unable to produce excess returns. In strong-form efficiency, prices reflect all historical, public and private information, and no one can earn excess returns. In addition, a similar perspective viewing the market is

the *random walk hypothesis* (Malkiel, 1973), which believes in an unpredictable prices series where all subsequent price movement is a random departure from previous prices.

2.2 Support Vector Machine

In machine learning, *support vector machine* (SVM) is a kernel-based learning algorithm introduced by Boser et al. (1992) and Vanik (1995). It was firstly applied on classification tasks and later adapted for regression task (Vapnik et al., 1996). Predominantly, support vector machine employs the "*kernel trick*" for projection of non-linear separable training data onto higher dimensional *feature space* by preserving dimensions of relatedness in the data. In a classification scenario it then obtains the maximum-margin hyperplane as the decision boundary pushed against by those *support vectors* and thus become capable of extracting the global optimal solutions regardless of the sparsity of the training data and less overfitted to it.

Given a set of training instances $\left(\mathbf{x}_i, y_i\right), i = 1,...,l$ where $\mathbf{x}_i \in R^n$ and $\mathbf{y} \in \left\{1,-1\right\}^l$, the support vector machine solves following optimisation problem:

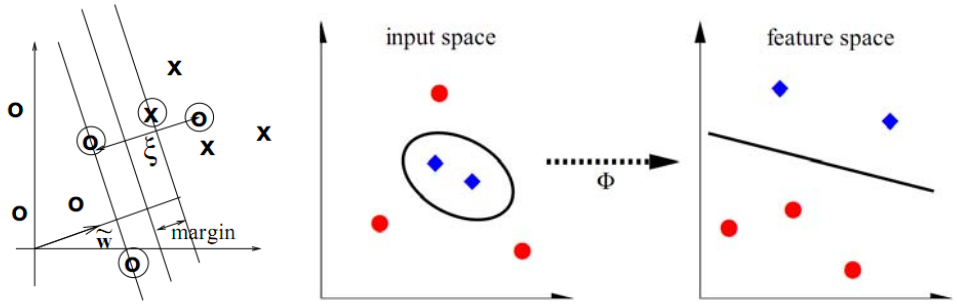
$$\begin{aligned} &\min_{w,b,\varepsilon} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ &\text{subject to} \quad y_i\left(\mathbf{w}^T\phi\left(\mathbf{x}_i\right)+b\right) \geq 1-\xi_i, \xi_i > 0 \end{aligned} \tag{2.1}$$

where ξ_i is the slack variable measuring the degree of errors and the function ϕ maps the training vector x_i into a higher or infinite dimensional feature space. Rather than compute the transformation in terms of dot product, the kernel function given as $K\left(\mathbf{x}_{i,}\mathbf{x}_j\right) \equiv \phi\left(\mathbf{x}_i\right)^T\phi\left(\mathbf{x}_j\right)$ provides an efficient way of doing the mapping to the higher dimensional space. In terms of classification tasks, the support vector machine maximises the margin of a linear classification boundary in the feature space. The optimal hyperplane is determined by only a few training data points which are called *support vectors*. Thus predictions for new inputs depend only on kernel function evaluated at a subset of the training data points. This makes support vector machine less overfitted to training set and become capable of tackling problems having very sparse data. A number of common kernel functions include

- Linear: $K\left(\mathbf{x}_{i,}\mathbf{x}_j\right) \equiv \mathbf{x}_i^T\mathbf{x}_j$
- Polynomial: $K\left(\mathbf{x}_{i,}\mathbf{x}_j\right) \equiv \left(\gamma\mathbf{x}_i^T\mathbf{x}_j + r\right)^d, \gamma > 0$

- Radial basis function $K\left(\mathbf{x}_i,\mathbf{x}_j\right)\equiv\exp\left(-\gamma\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2\right)^2,\gamma>0$

where γ, r and d are the parameters of the kernel functions.



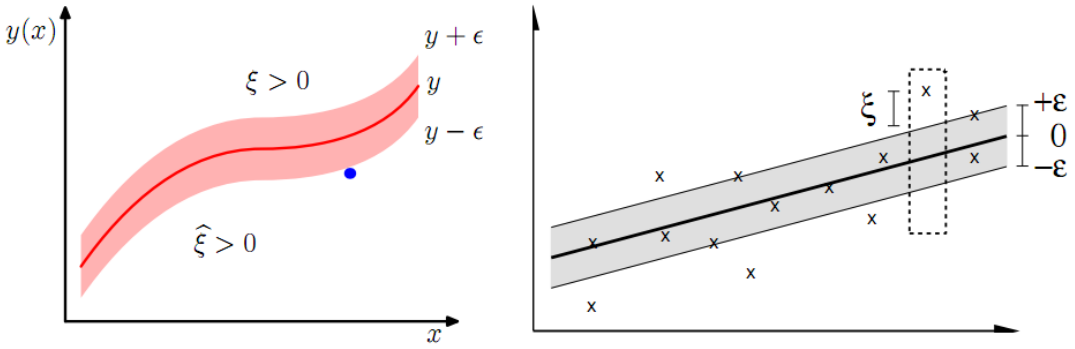
(a) (b)

Figure2.1: (a) A maximum margin classifier. (b) A kernel-based transformation

For regression tasks that predict real-valued output, *support vector regression* (SVR) differs from linear regression problem $y\left(\mathbf{x}\right)=\mathbf{w}.\mathbf{x}+w_0$ by obtaining sparse solutions which is addressed by the ε-insensitive error function (Vapnik, 1995), given by

$$E_\varepsilon=\begin{cases}\left|y\left(\mathbf{x}\right)-t\right|-\varepsilon & \text{if } \left|y\left(\mathbf{x}\right)-t\right|\geq\varepsilon;\\ 0, & \text{otherwise}\end{cases} \tag{2.2}$$

This function gives zero value if the absolute error between the prediction $y\left(\mathbf{x}\right)$ and the target t is less than ε where $\varepsilon > 0$. Hence, like the support vector machine for classification, the predictive model generated by support vector regression depends only on a subset of the training data as the ε-insensitive error function ignores data points that are close (within the threshold ε) to the predictions and only those lay outside of the ε-tube, which are allowed but penalised by the slack variables, decides the tube width:



(a) (b)

Figure2.2: (a) ε-insensitive SVR with polynomial kernel (fits one data point)
(b) A linear kernel SVR errors minimised with slack variables

In term of applications, support vector machine has been widely adopted including usage in text mining. As the nature of the textual data, sparsity may cause severe issues to some learning algorithms. As discussed earlier, the optimal solutions of support vector machine only depend on a subset of the training data, the support vectors, so that it handles the sparsity of the data well. Although the very sparse data in texts make the feature space more likely linear separable so that the kernel transformation may not improve the performance, it has been shown that support vector machine with a linear kernel (LSVM) outperformed most of the classifiers and reported by several works as the most accurate text classifiers (Yang and Liu, 1999).

More details of support vector machine can be found in Burges (1998) and Vapnik (1998) and Smola and Schölkopf (2004) for support vector regression

2.3 Feature Selection

Textual data have very sparse feature space. Feature selection is a common pre-processing step to restrict the dimension of the space to a subset of the features in order to reduce the computational complexity and improve the performance with less irrelevant attributes. There are generally two paradigms of feature selection methods: filters and wrappers.

Filter methods evaluate subsets of feature based on utility metrics that are independently of learning schemes. For example, in text classification problems filter methods compute a utility measure $A(t,c)$ between each feature t and the class c and rank the features based on the measure scores and select those exceeding an adjustable threshold. A wide range of statistical methods can be used as the utility measure, such as mutual information, x^2 test, or document frequency, etc.

On the other hand, *Wrapper* methods employ a learning scheme as a black box and evaluate subsets of features with respect to their predictive power. As the space of all combinations of features can be huge which make exhaustive search in the space a *NP-hard* problem (Amaldi and Kann, 1998) and becomes computationally intractable, one usually needs to decide the search strategies such as best-first, branch-and-bound, genetic algorithms, etc. The best first search strategy is the most common one which comes in two favours: forward selection and backward elimination. *Forward selection* starts with empty set and gradually adds the best fea-

ture to the set by evaluating on each of them where as *backward elimination* starts with the full feature set and progressively eliminates the least helpful ones. In general, wrapper methods are computation intensive but usually can lead to performance improvement. Sometimes it may be overfitted to the specific training set. By using some advanced search methods such as linear forward search (Gutlein et al., 2009) over the classical greedy or sequential best first search methods, wrappers can perform well on large scale feature space.

2.4 Chapter Summary

In this chapter, we introduced the key hypothesis behind stock market prediction tasks with respect to information efficiency and gave a general introduction to the techniques that we will utilise to build our models. Although efficient-market hypothesis treats accurate prediction of stock markets impossible in the long run, it believes that a market is correct in the sense that all prices always represent the best estimate of companies' future performance by aggregating all probabilities at the current time. The level of adjustment in future is due to arrival of new information and people's different reactions to it. Therefore, there exist correlations between unseen news and price movements which we intend to model. Support vector machine is a powerful technique to learn complex relationships in high dimensional feature space and feature selection methods has shown to improve accuracy for learning schemes in which we would like to explore its potential in our settings.

Chapter 3

Social Media

In the influential book *Warp Speed: America in the Age of the Mixed Media*, Kovach and Rosenstiel writes (1999):

> The classic function of journalism to sort out a true and reliable account of the day's events is being undermined. It is being displaced by the continuous news cycle, the growing power of sources over reporters, varying standards of journalism, and a fascination with inexpensive, polarizing argument. The press is also increasingly fixated on finding the 'big story' that will temporarily reassemble the now-fragmented mass audience.

We are in a "Smaller World" (Travers and Milgram, 1969) than before as the digital world makes information prosper and propagate faster. In recent years, this phenomenon has been even strengthened by a new form of media, namely, the social media. In this chapter we characterise this new type of content on the Web and more concentrate our discussion on one of main focuses of the media, blog and micro-blog, in order to set our main task in the context.

3.1 The Rising Genre

Nowadays, tens of millions of users are actively putting their thoughts, interaction with friends as well as their writing for self-expression and self-empowerment (Blood, 2002) onto the Web. The way information organised on the Web is getting transformed in someway from previous genres. Social media refers to a new form of media that gives attention to on user-generated content (UCG) created by accessible and scalable publishing tools which also provides explicit support for social interactions between users. Social media is commonly used as a personal

publishing and communication tool, where blogging, micro-blogging and social networking are the three main activities. Blog and micro-blog are a collection of pages, more specifically, author-oriented content, usually published through a template-based system and listed in a reverse chronological manner. It promotes interactions with readers and hyperlink-oriented communication with other authors. Although there is little technical secret behind the scene, the novel form of publishing and organising information and the large number of users involved make it considered by some researchers as a new form of digital communication genre and studied from multiple perspectives (Herring et al. 2004) including tasks like profiling, community analysis or new ways of searching. Since the contents in most blogs are provided by one or two authors exclusively, building profiles of the blog can reveal certain characteristics of the authors. Profiling can be utilised in a wide range of purposes including commercial ones such as advertising and promotion as well as personal aspects like topical opinion exchange and latent friend discovery (Shen et al. 2006).

Bloggers form communities based on interests and interact with each other. Discovering the interrelationship and interaction between bloggers is an extension to the social network analysis. The relationship in the digital social space can be expressed either through explicit hyperlinks or through implicit semantic relationship from the generated content. The authority and hub effects also exist on the social map where information sources and seekers stand for different user intensions. As research tasks conducted in the TREC 2006 and 2007 Blog Track, Ounis et al. (2007) that the subjective nature is a key feature that distinguishes the user-generated contents from the factual content used in other text retrieval tasks such as Web search. Many retrieval queries in such context are of named entities such as person names. The underlying information needs seem to be of opinion-oriented or perspective-finding aspects.

The following figures show sample interfaces of two blog and micro-blog sites:

(a) (b)

Figure3.3: (a) A blog front page; (b) A Twitter profile page

The following figure demonstrates a social map of a "game" community formed spontaneously on the micro-blogging site Twitter:

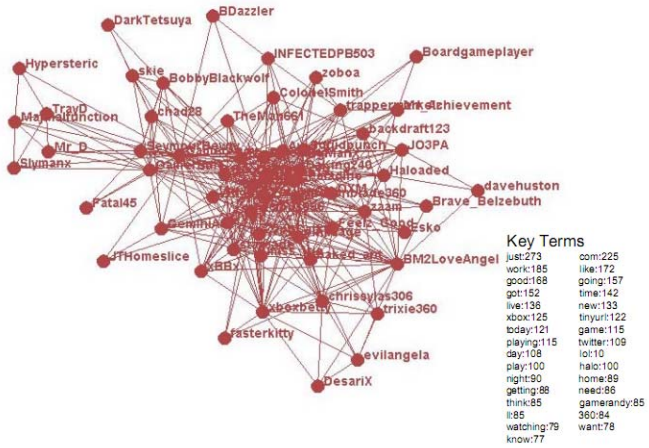


Figure3.4: a micro-blogging community sharing common "gaming" interests (Java and Song, 2007)

3.2 Twitter, Micro-blogging, and Real-time Search

Twitter, an online micro-blogging service created in 2006, allows users to publish their own words in less than 140 characters each time and share with others. Users can access the service from multiple ways, via the official website, Short Message Service (SMS) or external

applications. As its name suggests, the most distinctive feature of the micro-blogging platform is the emphasis on real-time update and explicit support of interaction between users. The only relationship between users is follow-and-be-followed where users subscribe to each other and keep up-to-date with their updates simultaneously. In the following figure according to Alexa[1], Twitter is gaining increasing popularity since 2009:

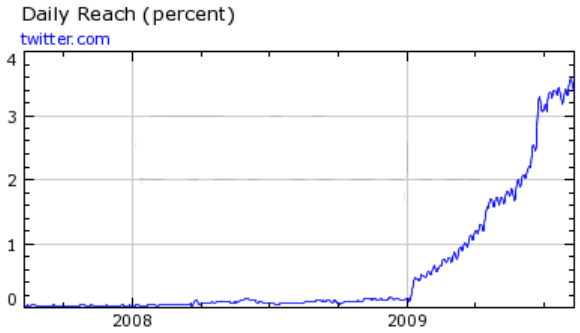


Figure3.5: Daily reach of Twitter measured in terms of the percentage of global Internet users

Another distinct feature of Twitter is the real-time and multi-faceted updating mechanisms which enable very rapid release of news information. In a recent work conducted on 90 million news and blog articles over 3 months time, Leskovec, Backstrom and Kleinberg (2009) identified that most of the early sources of news are independent media and blogs well ahead of large media organisations. As mentioned earlier by Kovach and Rosenstiel (1999), the traditional way of news propagation cycle is decentralised by the heterogeneous media in the digital world. However, in the recent development of blogosphere, it has been seen that many influential bloggers and news organisations tend to have twitter profiles synchronised and updated at the same time with the hope to reaching its audience in seconds. This enables real-time reflection of the news world on a single site where Twitter plays as the centralised point of all updates. In addition, the crowd of other users can also provide valuable sources of news by their personal experience and timely updates, as seen in events like the 2009 US Airways Flight 1549 incident and the 2009 Iranian presidential election protests. These recent changes spawn another usage of the social media, the *real time search*, which, comparing to the traditional index-and-rank based search engines, concentrates on channelling information updates from multiple sources, usually social media sites, to bring users a timely perspective of their interests. According to Mishne and Rijke (2006), there are generally two types of information on the Web: factoid and opinionated, where social media tends to have the latter one and people seek news-related que-

[1] http://alexa.com

ries in the social space rather than very detailed information needs such as factoid questions asked to a general search engine. Therefore, the divergence of usage and development of the traditional Web and the social media continues.

3.3 Chapter Summary

In this chapter, we introduced a new form of media and content on the Web, the social media and user-generated content. The characteristics of the blog and micro-blog space bring new shapes to the digital world. We specifically described a popular micro-blogging service, Twitter. Its simple form of updating and sharing and real-time reflection on the news worlds provide us an accurate data sources for our experiments.

Chapter 4

Experimental Framework

This chapter outlines various experimental settings. The conceptual designs of the experiments come from a number of decisions which are also discussed. We firstly introduce the main data-set and put the emphasis on a hybrid approach for feature designs which includes a number of filter and wrapper selection methods. Evaluation methods are discussed at last.

3.4 Choice of Dataset

We collect three types of data: Newswire, Blog and Twitter. News data is the traditional and most common published material on the Web. It conveys factoid information, usually edited to include point of views from editors and reporters. In the financial markets, private and public news usually have significant impact on stock prices as traders rely on them to make judgement for future trading decisions. Modern financial trading systems integrate with quick-responding news monitoring and alerting functions in order to bring timely advantage; whereas the general public mostly obtain news through mainstream media organisations. These organisations are the central points in news propagation cycle but also maintain a reasonable lag in publication after the first release of news (Leskovec et al., 2009). Most previous stock price forecasting tasks based on textual information utilise news data, usually the headlines (Lavrenko et al., 2000; Fung et al., 2003). Headlines are the most influential to the general public and to stock prices, however, the reception and reaction to them differ significantly from readers to readers and time to time. Since releasing intervals and companies or industries mentioned in headlines cannot be defined systematically, it is insufficient to build a prediction model for a specific company purely based on headlines.

As mentioned, Twitter is a good source of information due to timely updates by its increasing user base from various backgrounds. The simple form of publishing contains less noise data comparing to heterogeneous news web pages with large amount of advertisement.

The follow-and-be-followed ties on the Twitter social map make information spread timely and efficiently. In addition, we find that the length restriction of the posts makes users to write efficiently with meaningful keywords. This increases the density of useful information and those keywords are more likely to be repeated by other users. This can bring benefits to text processing tasks as word frequencies increase.

Comparing to blogs, Twitter has higher author/reader ratio due to easier access to publishing. As examined by Java and Song (2007), the user activities on Twitter fall into four categories: daily chatter, conversations, sharing information/URLs and reporting news. Majority of Twitter posts fall into daily chatter category where people put answers to the top question on site: "what are you doing". According to our observation, this source of information only contributes to a small portion of data in the blog dataset. On blogs, bloggers, especially those influential ones, tend to put more serious, thoughtful and well-organised opinions rather than simple expression of daily routine or transitory feeling about a topic. Such intention of writing also opposes the other three common uses of Twitter as it discourages short and plain postings. Therefore, Twitter data provide a richer and abroad source of information for our experiments over blogs.

In light of above justifications, we set our experiments mainly based on the Twitter dataset which consists of 61,756,056 posts crawled since February 2009 to June 2009. The daily amount of crawled posts is shown in the following graph:

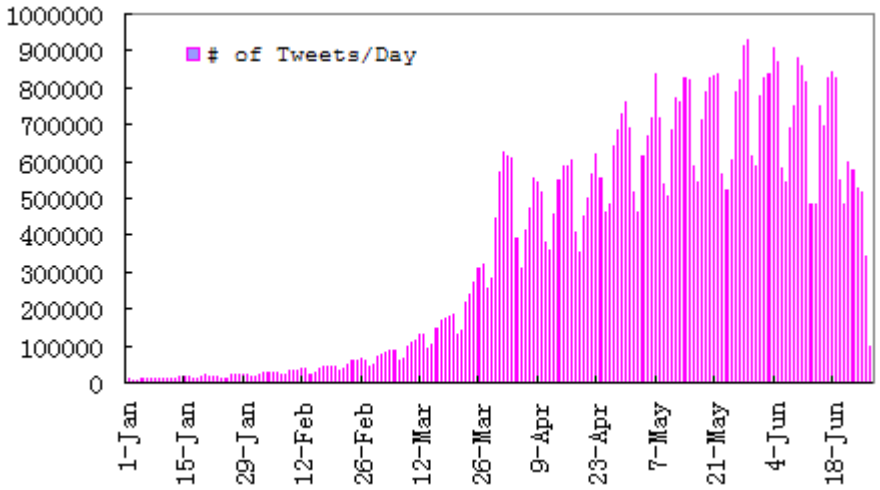


Figure4.1 Number of Twitter posts crawled per day

As can be seen, the amount of crawled pages increases quickly till April to reach the level of 500,000 posts per day and peaked in May and June for over 800,000 posts per day. In order to obtain sufficient sources of data, we use data from April to June for our prediction task.

3.5 Experiment Design

The aim of the project is to determine whether and to what extent textual data extracted from social media helps in predicting stock prices. As there is little previous work implementing similar goals, we set up our experiments to cover a wide range of settings in order to obtain comparable results. Our perspective has an emphasis on feature designs rather than tuning on algorithm parameters. The best performed learning parameters can be estimated through standard cross-validation on given training sets but the performance and generalisation ability of the model still rely on the input feature variables that bring good predictive power on stock prices. In the following section, various parts of experiment design are outlined.

4.1.1 Stock Selection

As reported by Mishne and Rijke (2006), people are more engaged in political and technology related information on social media. Therefore, to obtain a sufficient source of relevant data, we firstly choose three technology companies as our predicting targets: Yahoo! Inc. (YHOO), Google Inc. (GOOG) and Microsoft Corporation (MSFT), in which we set Yahoo! Inc. as our main target for prediction and analysis. Yahoo! Inc. has wide range of coverage in the dataset but not the most popular one so that the experiments carry more generality. We are also interested in seeing how overall index can be predicted therefore the NASDAQ Composite, S&P 500 and FTSE 100 indices are included. Furthermore for comparison purpose we also choose HBSC Holding plc (HBC) which has less mentioning on the Twitter platform. For each stock the closing price on each day is chosen.

4.1.2 Feature Filtering – Noun phrasing, Loose n-gram model and Document frequency thresholding

Bag of words is a common feature representation in text mining tasks where unique words are put into an unordered "bag", disregarding grammar and word order. However, putting all words into the bag leads to sparse feature space and considerably increases computational complexity but may drop performance comparing to using only a subset of more useful terms (Yang and Liu, 1999). In the Web domain, the number of terms in a vocabulary increases sharply where most of them contribute little meaning to machine learning tasks. Feature selection plays an important role in this context. Yang and Pedersen (1997) show in their benchmark work that filter techniques like information gain and x^2-test discussed in the previous chapter can remove upon 90% of unique terms and actually yield improvement in accuracy in classification tasks. As examined in the Twitter dataset, there are 6,159,239 unique word terms, i.e. strings of

non-symbolic characters. Such sparse data imposes high computational cost as well as the risk of overfitting. In the experiments we design a hybrid feature selection method to address such problem.

According to our observation shorter posts in Twitter dataset make the topics change more frequently. Within the 61 million posts there are only a few topics that are related to the stock market and to the stocks selected for prediction. A simple way to distinguish such topics from other unrelated ones is to use the *noun phrasing* (Tolle and Chen, 2000). This method forms compound noun phrases in order to capture a "richer linguistic representation" (Anick and Vaithyanathan, 1997) comparing to bag of words representation where single terms lack an appropriate level of context and thus fail to reflect more complex semantic concepts. In the experiments, we take noun phrasing as the first filtering step. We identify nouns in a Twitter post and keep the post for further processing only if it explicitly mentions the company name of the target stock; or for composite indices, the name of any company that is included in the index.

In order to choose other words in the noun phrases we make use of a model introduced by Zhang and Zhu (2007), the *Loose N-gram* features, defined as co-occurring words within limited range by disregarding word ordering. The model has the benefit of traditional n-gram model that retains useful contextual and semantic information. Their study shows improvement of such feature over unigram model in text classification tasks on Reuters-21578 and TREC-2005 TC datasets. In our context, as the Twitter posts have a fixed maximum length, the loose N-gram model can be applied naturally to form noun phrases where the company name of the stocks are combined with co-occurring words in the posts. In the experiments, we build loose trigram and loose bigram models.

Further, we take two approaches in term of selecting the co-occurring words. The first method keeps all individual terms and combines them with the noun; the second one based on the ideas of topic filtering (Li et al., 2005) extracts only terms that are business-related which are filtered by information gain algorithm trained on a tagged version of Twitter dataset. The dataset contains documents that are labelled as "Business", which are Twitter posts that contain business-related tags, versus "Non-business" ones. We explicitly find tags in the posts as information gain algorithm works on a category-basis and our regression targets are real-valued which provide no explicit category labels. The top 20 significant terms identified by the information gain algorithm are illustrated below:

#	Term	Significance	#	Term	Significance
1	finance	0.04485	11	companies	0.02834
2	marketing	0.04456	12	market	0.02804

3	Business	0.04426	13	sales	0.02776
4	Investing	0.03998	14	entrepreneurs	0.02759
5	investment	0.03809	15	markets	0.02388
6	Financial	0.0351	16	invest	0.02369
7	management	0.03448	17	estate	0.0235
8	businesses	0.03115	18	stocks	0.0233
9	Investors	0.03	19	company	0.02275
10	entrepreneur	0.02857	20	investments	0.02261

Table4.1 20 top-ranked significant words in classifying business Twitter posts

According to our pre-experiment test, without topic filtering the loose bigram model still generates 53,673 unique features for Google's stock and 36,173 features for Yahoo!'s stock; whereas the loose trigram model generates even more. Although evaluation will be done in terms of the effectiveness of previous filtering approaches, there is still room for further selecting the features. As another filter method, *document frequency thresholding* is simple but effective as observed by Yang and Pedersen (1997), where they put the following statement with respect to the term frequency:

> The excellent performance of DF, IG, and CHI indicates that common terms are indeed informative for text categorization tasks. If significant amounts of information were lost at high levels (e.g. 98%) of vocabulary reduction it would not be possible for kNN or LLSF to have improved categorization performance. To be more precise, in theory, IG measures the number of bits of information obtained by knowing the presence or absence of a term in a document. The strong DF-IG correlations means that common terms are often informative, and vice versa.

Therefore, in the experiments we further set up various thresholds to filter the loose n-gram features and compare their usefulness.

4.1.3 Feature Selection – A Hybrid Wrapper Method

Within feature selection domain, wrappers are reported to generate better performance over filter methods (Chiang et al., 1996; Lawrence et al., 1997; Min and Lee, 2005; Gutlein et al., 2009). However, its high computational complexity restricted its applications in many areas. As introduced by Liu and Zheng (2004) and applied by Lee (2009) in stock market prediction, a hybrid feature selection method, which combines filter and wrapper methods by filtering out some "non-informative" features before submitting to the wrapper part, can reduce the number of model to be evaluated and thus improve its efficiency. Lee's work reports significant im-

provements in efficiency as well as effectiveness over conventional wrapper methods in stock trend prediction scenario which employs technical indicators as original features. Our experiments also evaluate this hybrid feature selection approach but apply on the textual data. In previous stages, we build features upon noun phrasing, loose n-gram model and document frequency thresholding and an optional topic filtering step. Next, a wrapper subset selector is applied to the filtered feature space by wrapping our target support vector regression algorithm and internal cross-validation. The final set of selected features is due for training and evaluation on a third test set. This process is illustrated in the [Figure]

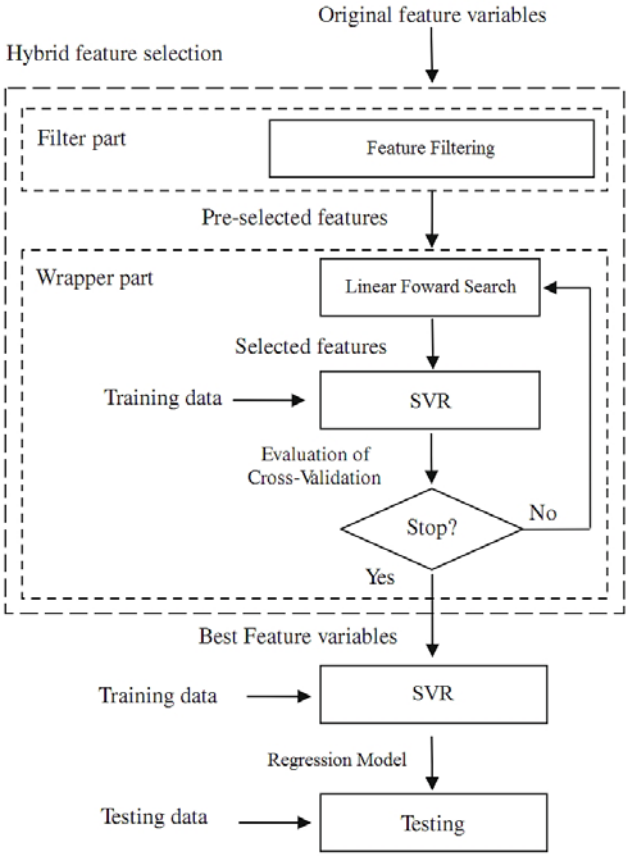


Figure4.2 Hybrid feature selection procedures (Lee, 2009)

As discussed in Chapter 2, wrapper methods require a search algorithm to decide the best features in each step. In terms of classical forward search strategies, a frequently used implementation is Sequential Forward Selection (SFS) which performs a simple hill-climbing search. Although forward search strategies maintain a computational advantage over backward elimination as the starting subset contains fewer features, the computational cost is still high as the number of learning models to build and evaluations in each iteration equal to the number of features remained for selection, which grows quadratically with respect the size of the whole

features space. Gutlein et al. (2009) recently introduced an advanced forward search algorithm called Linear Forward Search to address the runtime issue as well as the common overfitting problem using wrappers. The algorithm reduces the number of subset evaluation in each selection step by precomputing a threshold k so that the level of search expansion is limited to linear complexity. Their work reports a notable reduction in runtime without losing accuracy comparing to a full forward search. In our experiments, we are interested to apply this approach to the feature set and evaluate its effectiveness.

4.1.4 Noun Phrase Expansion

In information retrieval, query expansion can lead to improved performance (Abberley et al., 1999). As introduced, we choose the noun phrasing approach which can be treated as submitting queries to retrieve the features set regarding to the company's information that correlates with stock prices. Thus, we are interested in finding out whether expanding the querying nouns to retrieve more related company's information also improves the performance and accuracy. In the experiments, we will experiment expansion on several other companies with different level of relatedness. The effectiveness are evaluated and compared to with the unexpanded set of model performances.

4.1.5 Learning Algorithm

In order to provide a common ground for various methods we employed in the feature preparation stage, the configurations of learning algorithm remain constant in most experiments. As mentioned in the previous chapter, we use a ε-insensitive support vector regression algorithm with a linear kernel. The value of ε is set to .001 which provides reasonable performance in most tasks. And linear kernel performs well in text-based tasks as reported by several works (Yang and Liu, 1999; Dumais and Chen, 2000) as the sparse data is mostly linearly separable. Our pre-experiment test shows similar results on the linear kernel comparing to RBF kernel and it also maintains a better efficiency in computation.

4.1.6 Sample Window and Other Setting

Our task is by taking the past n days of data as training samples to predict the prices on $(n+1)$th day. In order to obtain comparable results, we also evaluate a number of different values of n. In addition, for all our experiments, other common settings include standard stop words removal and term-counting as feature values with no explicit weighting scheme applied.

3.6 Evaluation Method

In technical analysis, *moving average* is an important and reliable statistical indicator. It helps traders to track the long-term trend of a stock by automatically incorporating historical pricing data into the model so that short-term noisy fluctuations are smoothed out. A common unweighted moving average model is the *simple moving average* (SMA) defined as:

$$\mathrm{SMA}_t = \frac{1}{p}\sum_{i=1}^{p} X_{t-i} \tag{4.1}$$

where t is the value on target date and p is the model parameter indicating the number of previous instances taken into account. In our experiments, we take p=3, i.e. 3 days moving-average, as the baseline method. In addition, the moving averages inside a same sample window as the comparing method are also feed into the support vector regression model to obtain predictive results.

In order to compare results with the baseline, mean squared error (MSE) defined as the difference between the actual observed value and the value predicted by the model is used, more specifically:

$$\mathrm{MSE}(\mathbf{t}) = \frac{1}{N}\sum_{i=1}^{N}\left(y(x_i) - t_i\right)^2 \tag{4.2}$$

where t is the observed values, y is the predictor and x is input data points has a dimension of N. We utilise the root mean squared error (RMSE) which is taken as square root of MSE in order to keep the value the same unit as the quantity being estimated, i.e. stock price in U.S. dollar or pound sterling. To evaluate the overall performance of a chosen method we build a series of 1-day lag prediction models over the three months data, i.e. from the first day we use data prior to that day to build a predictive model for it and move forward until the end of three months.

On the other hand, although mean squared error measures the overall "closeness" of model performance according to the true price trends, it fails to reflect the "swiftness" a model reacts on turning points in a price series where the trends turn to an opposite direction. As far as prediction tasks concern, continuous lags in such swift response and accuracy make a prediction model useless. Moving average is a lagging indicator where it constantly lags behind the current price and only reflects the "historical prices". A lower mean squared error than the moving average does not mean the model is accurate. Therefore, we aim at measuring our prediction models in terms of such criterion where we align the curve generated by our model with the true price trends and see how synchronised the two curves are. A curve with less lagging and act simultaneously on turning points are treated as more accurate.

3.7 Chapter Summary

This chapter discussed the methods and conceptual design of the experiments. We firstly provided a justification on our choice of experimental dataset and described several important aspects of experiments including selection of stocks, feature filtering and selection methods, learning algorithm and evaluation details.

Chapter 5

Experiment Evaluation

This chapter describes a number of experiments we conduced according to the previous discussed methods. Along with the results shown in each section, critical evaluations and analyses are carried out in order to provide a coherent conclusion on each method. There are a number of settings vary across the experiments. In the next section we introduce a set of naming conventions to distinguish between the different settings. Then we start our experiments in the simplest setting in order to motivate more complex methods, for which we can study the improvement or drop in performance where our final findings are drawn from.

5.1 Naming Convention

In the experiments, we introduce the following naming conventions to be used as suffixes in order to denote a number of variable settings. The names of all prediction models start with "Pred" followed by the respective configurable suffixes.

- bi / tri / slf indicates whether the noun phrases used as features are formed by the loose bigram model, trigram model or only counted on its own.
- biz indicates the features are filtered according to the business terms computed according to information gain criterion. When missing, no term is filtered out with respect to their topics. According to our pre-experiment testing, we choose the 400 top-ranked terms. A full list of the terms and their significance is given in the appendix section.
- df_n indicates the document frequency threshold used to filter terms. When set, terms occurring less than *n* times in the training set are discarded; otherwise all occurred terms are kept.

- sp_n indicates the number of samples included in the training set. Each training sample is a vector containing previous day feature values along with price on the target day. In the experiments we vary a number of different n = 3, 5, 7, 10, 20.
- fs indicates whether the wrapper method for feature selection is applied.
- $big3$ / $inet$ / $trad$ indicates type of expansion into the base noun set.

5.2 Simple Noun Counting

We start our experiment at the simplest setting. We only use one feature at the time: the number of mentions of the target company on each day. Although we will mainly focus on Yahoo! stock in later sections, now we introduce two stocks for comparison purpose. The root mean squared error values for the two stocks are reported in Table 5.1.

Setting	RMSE	Baseline	Difference
Yhoo-slf_df1_sp3	0.550	0.464	-18.53%
Goog-slf_df1_sp3	8.618	7.846	-9.84%

Table 5.1: RMSE values on simple counting methods

As can be seen, the baseline of 3 days moving averages performs better than the models. Figure 5.1 illustrates the model performance during the 3 months period where the predicted prices for each day are plotted against the true prices and the moving average.

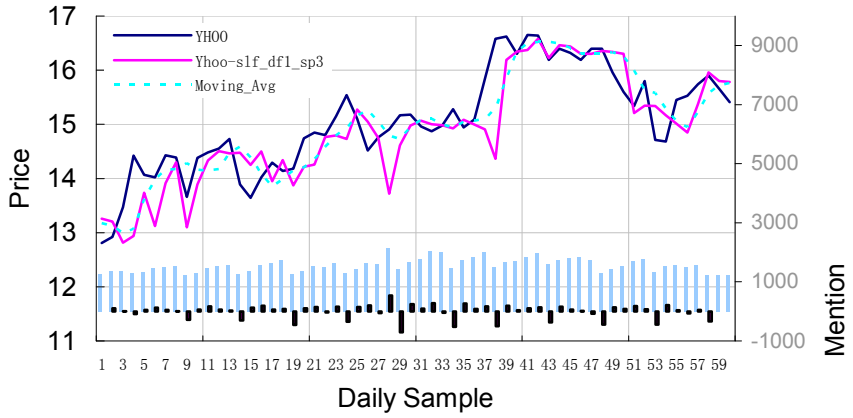


Figure 5.1: Fitted curve in setting “Yhoo-slf_df1_sp3”

In Figure 5.1, the blue bars at the bottom indicate the number of mentions of the company

on each day, i.e. the feature variable fed into the model; and the dark bars indicate the difference in the value of mention comparing to the previous day. It is interesting to see that the simple counting method do bring a reasonable fitted curve to the actual prices. It performs slightly worse than the baseline as can be seen that there are the some "unbalanced points", such as day 28 and 39, where the prices are raising but the predicted values plunge to the opposite direction so that errors increase sharply. When look at the actual number of mentions under these points, it can be observed that there are sharp changes in terms of the number of mentions on these days from the previous day. Therefore, the model is very sensitive to such change in value even if these changes do not bring a significant impact on price. This is true as our model parameters are estimated solely on this variable at the time. However, if it is the case for the change in mentioning to have an impact on the real prices, such sensitivity makes the model react to the price changes swiftly, without a lag that is constantly shown on the moving average curve. For instance, on day 3 to day 10, the true price fluctuates for a number of times and the moving average curve fails to react on these changes since all short term fluctuations are automatically smoothed out. On the other hand, the simple counting model does reflect these movements in time. Additionally, Figure 5.2 plots the fitted curve for Google Inc. Apparently, the curve has less "unbalanced points" and is nearly identical to the moving average curve except at only a few points. Such stability reveals an important aspect of our modelling and our intention for applying the simple counting method. Although we have only one simple variable, as we feed the model with adjacent 3 days historical data, similar to the moving average model, our model maintains a memory of such history by obtaining the regression parameters and the intercept through training. Such memory defines a baseline performance with a lagged prediction as long as there is no dramatic change in variable values upon prediction. In order to obtain simultaneous response on turning points in price series, we need to define a set of feature variables that can reveal the true relationships in the price movement and their values to some extend, as least at the point of training. As it has been observed, although the feature variable of simple noun counting method maintains such correlation at some points, it fails to capture the relationship solely and constantly.

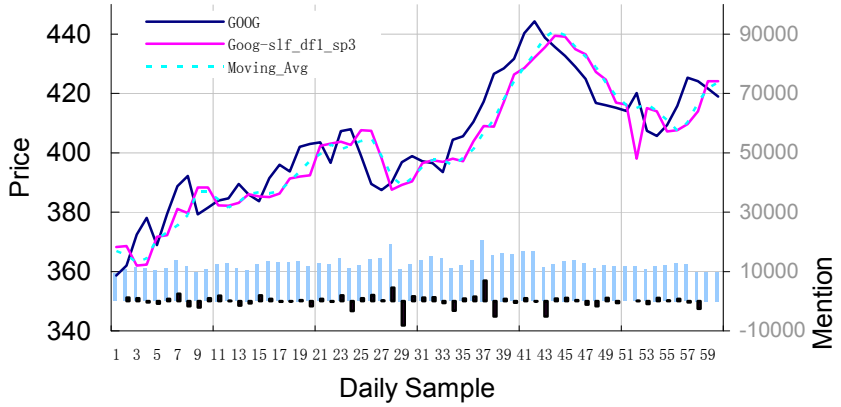


Figure 5.2: Fitted curve in setting “Goog-slf_df1_sp3”

Training Sample Size and Kernel Function

In order to show the effect of different previous days samples included in training set, we vary the value of n=20, 12, 7, 5, 3, 2, 1. Figure 5.3 gives the results where the error decreases along with the number of training samples until n=2 where we notice a very large error. As shown in Figure 5.4 comparing n=2 and n=3, we find the *sp2* model gives an extreme prediction of 5.85 for target value 14.42. According to our observation, the data points in the sample training set have similar value thus lie very close in space. Due to limitation in the linear kernel used in the regression algorithm where close and limited number of training points cannot provide necessary “support” to the algorithm, the intercept is assigned a large negative value. The error is back to 0.447 when a RBF kernel is applied which transforms the data points to a higher dimensional space. Therefore, for our future tasks to obtain a better performance and avoid unnecessary confusion in algorithms, we set n=3 accordingly.

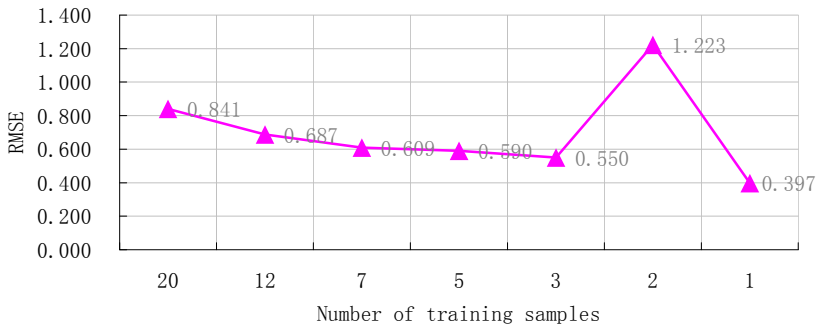


Figure 5.3: RMSE with varying sample size

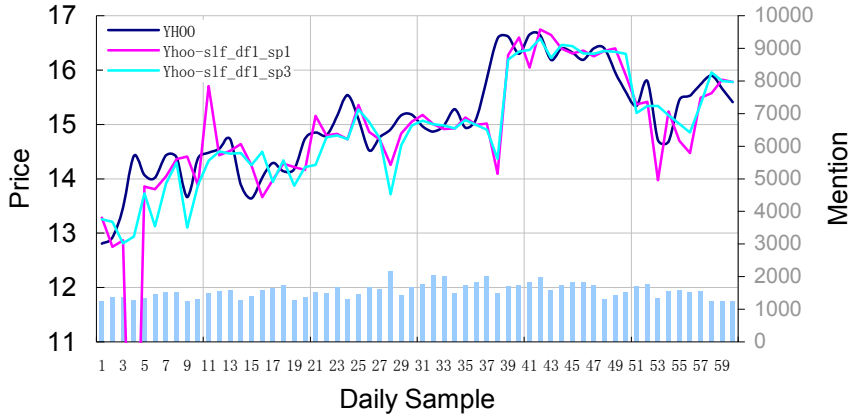


Figure 5.4: Fitted curves when sample size *n*=2 and 3

5.3 Loose N-gram Model and Topic Filtering

In order to investigate the effects of different loose n-gram models, in this section we show the experiment results from models built upon features in either loose bigram or trigram models with varying document frequency thresholds. The results shown in Table 5.2 are divided into four sections where the top two employ topic filtering in either a bigram or trigram model and the bottom two has the same division but without filtering features according to the business topic.

As it can be observed in Table 5.2, the significant improvement over the simple noun counting method is that all four categories of methods can lead to performance better than the moving average baseline. Methods with and without topic filtering have little difference in terms of RMSE. However, one should note that topic filtering can bring different features sets. With topic-filtered set, as the maximum number of features is fixed, although it is able to capture more business-related terms, it cannot find out some popular terms like “bing”, “iphone”, “geocities” etc. which may bring more useful information. On the other hand, methods without topic filtering can bring noisy terms like “rt” which is an extremely high occurring term on Twitter when people “re-tweet” other people’s posts. In addition, it can be observed that smaller frequency thresholds increase the number of selected features and can bring significant improvement on performance. This can be viewed as the expanded feature set does bring more information. However, when the feature set is so large that it contains more noisy terms, the performance drops. For example, it has been shown that performance decreases in setting “*Yhoo-bi_df5_sp3*” with threshold set to 5 which has 3,133 features comparing the threshold of

10 in which has only 774 features.

In addition, the difference in terms of RMSE between bigram and trigram model methods is rather small where trigram model methods maintain a marginal advantage over the bigram ones. When we come to the fitted curves generated by the models, the difference between methods arises. As shown in Figure 5.7 and Figure 5.8, with topic filtering, the models have nearly identical curves as the moving average ones in which they maintain a constant lagging in responding sharp changes in price trends. In contrast, models without topic filtering as shown in Figure 5.5 and Figure 5.6 respond to these turning points at the same time as the true price trends, for instance, on day 9, 22 and 54. Although the predicted changes at these turning points may not be sharp enough, they are still proof of improvement for the models to perform more swiftly and accurate on prediction comparing to the lagging baselines.

Furthermore, comparing bigram curve in Figure 5.6, the trigram curve shown in Figure 5.5 matches the underlying prices better, for instance, on day 10 and 21. However, it still suffers in lagging at several points. The histogram at the bottom of the RMSE values shows that the large errors, for instance on day 4 and 38, mainly come from lagging when the actual price trends up rapidly but the model curve follows slowly with a large difference in value on the same day.

To summarise, by introducing more features into the noun phrases using loose bigram and trigrams with a lowered frequency threshold, more information is brought into the model and the performances increase over the baseline. With topic filtering methods, the models bring smooth curves that carry lag similar to those in baseline; whereas in methods without topic filtering, there are naturally rising terms captured and correlations between these terms and the true prices built into the models can improve the accurateness of the models to be synchronised in detecting several turning points in the price series.

Setting	RMSE	Baseline	No. Feature	Top Feature
Yhoo-bi_biz_df40_sp3	0.670	0.464	13	social,music,ceo, traffic,web,email,data, finance, jobs, money, client, marketing, networking, analytics, business, web, photo
Yhoo-bi_biz_df20_sp3	0.582	0.464	31	
Yhoo-bi_biz_df10_sp3	0.474	0.464	79	
Yhoo-bi_biz_df5_sp3	0.454	0.464	164	
Yhoo-bi_biz_df1_sp3	**0.453**	0.464	358	
Yhoo-tri_biz_df20_sp3	0.711	0.464	30	deal_ceo, bartz_ceo, asked_site, search_web, travel_photo, rumor_site, google_traffic, search_data, carol_ceo, offers_free, networking_social
Yhoo-tri_biz_df10_sp3	0.476	0.464	275	
Yhoo-tri_biz_df5_sp3	**0.443**	0.464	1201	
Yhoo-tri_biz_df1_sp3	0.453	0.464	18761	
Yhoo-bi_df80_sp3	0.604	0.464	17	sideline, meme, gmail,

Yhoo-bi_df40_sp3	0.481	0.464	60	google, leapfrog, bing, search, geocities, mail, twitter, ceo, swine, Michael iphone, flickr, blog
Yhoo-bi_df20_sp3	0.474	0.464	241	
Yhoo-bi_df10_sp3	**0.447**	0.464	774	
Yhoo-bi_df5_sp3	0.455	0.464	3133	
Yhoo-tri_df60_sp3	0.631	0.464	25	loopholes_google, engine_bing, geocities_plug, flu_swine, bing_google, search_leapfrog, bartz_carol_yahoo, hamas_fatah_yahoo, nearing_facebook_yahoo
Yhoo-tri_df40_sp3	0.546	0.464	59	
Yhoo-tri_df20_sp3	0.464	0.464	603	
Yhoo-tri_df10_sp3	**0.441**	0.464	2544	
Yhoo-tri_df5_sp3	0.453	0.464	12580	

Table 5.2: Experiment results showing models in loose bigram and trigram model with and without topic filtering applied

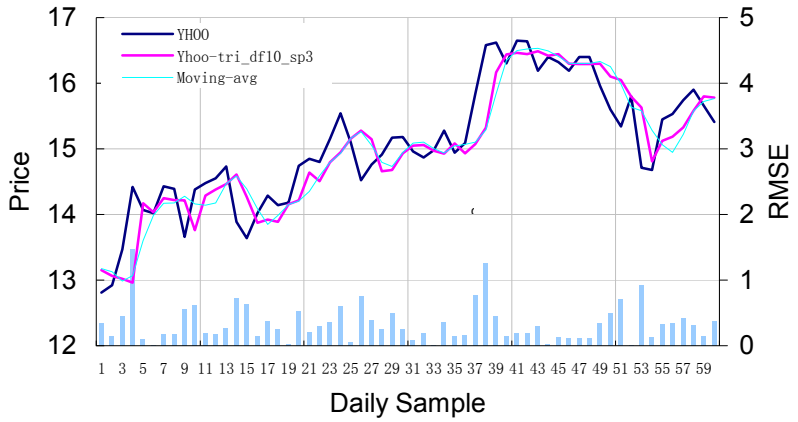


Figure 5.5: Fitted curve from trigram model, setting “Yhoo-tri_df10_sp3”

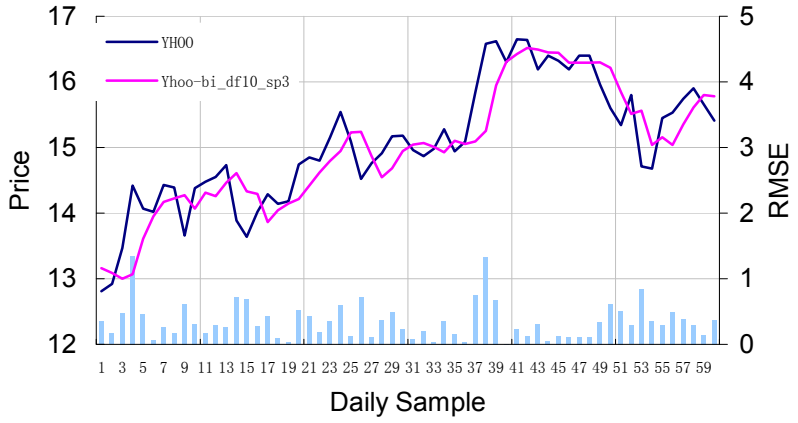


Figure 5.6: Fitted curve from bigram model, setting “Yhoo-bi_df10_sp3”

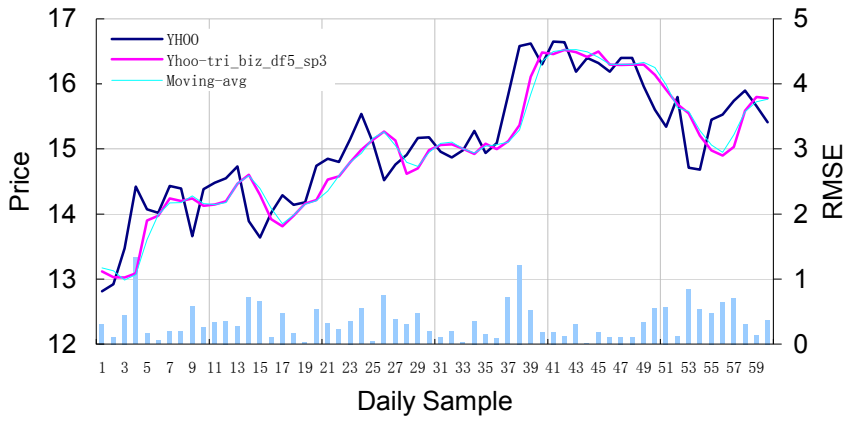


Figure 5.7: Fitted curve from trigram model with topic filtering, setting “Yhoo-bi_biz_df5_sp3”

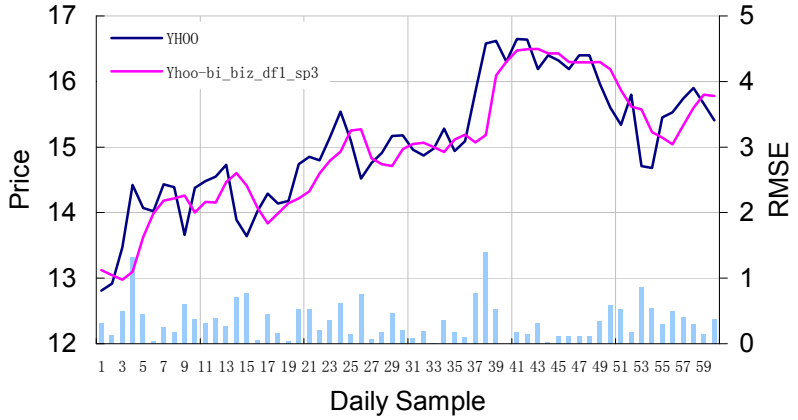


Figure 5.8: Fitted curve from bigram model with topic filtering, setting “Yhoo-bi_giz_df1_sp3”

5.4 Wrapper Method for Feature Selection

Based on the results from the last section, we are interested in finding out the effect of further feature selection using wrapper methods. As mentioned, since the three months data are divided into 60 training sets, we apply the wrapper algorithm on each training set to select out set-specific features and evaluate it on the hold-out test set which aims at the next day’s price.

As shown in Table 5.3, the overall performances measured by RMSE drop after applying feature selection in which only one setting has reached the same level as the baseline and the others are only getting close. However, it is insufficient to conclude the usefulness of the wrapper selection method as experiments in the previous section show the low error can actu-

ally come from "smoothness" but not "accurateness" where a number of turning points are either missed out or suffer in lagging. Comparing the selected top features to those only filtered by frequency in the last section, we can see that the difference is rather small, i.e. there are quite a few top features from previous experiments are selected by the wrapper algorithm again. As studied in the previous chapter, Yang and Pedersen (1997) state that filter feature selection methods like document frequency thresholding, information gain and x^2 statistics favour frequent terms and those terms are often informative. In our experiments, the intersection of selected feature set between wrapper and filter methods suggests that the wrapper method also favours common terms which are often informative and lead to more help in learning algorithms that are evaluated internally by the wrappers. As a result of selecting more informative features and drop other ones that might be noises, the model performances are expected to improve. However, we find out that with the current setting of linear forward search algorithm, the wrapper method only selects a very small number of features before it terminates. In all of the wrapper experiments we conducted which include evaluations on 1,080 training sets, there were only 2,697 features selected in total, meaning that on average there were only 2 features in each of the training set after applied the wrapper selection. According to our observation there were many of them actually have no feature. Such small or empty set of features would attribute to the drop in performance. However, when varying the parameter *k* of linear forward search algorithm in order to evaluate on more features, the runtime cost increases intensively which prevented us conducting a full series of 60 evaluations within the project timeframe. Such scenario confirms the restriction on the wrapper method where computational cost is the main reason preventing its wide adoption. Therefore, in order to conclude the overall performance of the wrapper methods, further exploration in the settings of linear forward search should be done.

Nevertheless, the effect of selecting more informative features by dropping noisy ones has an impact on the fitted curves. From Figure 5.9, we can see the curve fitted by the model with bigram features and without topic filtering is more accurate in prediction comparing to the original curve in Figure 5.4. At several turning points, for instances on day 4, 16 and 27, it responds to the changes swiftly and sufficiently when the original curve fails to capture the changes to such level. The similar effects do arise on the other three figures comparing to their original curves before applying wrapper method for selection. Consequently, such "better fitted" curve is a positive sign for the wrapper selection method.

Setting	RMSE	Baseline	Top Feature
Yhoo-bi_biz_df1_sp3_fs	**0.495**	0.464	music, social, data, web, email, traffic, ceo, seo, pay, photo, analytics, networking, marketing, free, money, account, site, jobs, internet
Yhoo-bi_biz_df5_sp3_fs	0.638	0.464	
Yhoo-bi_biz_df10_sp3_fs	0.558	0.464	
Yhoo-bi_biz_df20_sp3_fs	0.598	0.464	
Yhoo-bi_biz_df40_sp3_fs	0.790	0.464	
Yhoo-tri_biz_df1_sp3_fs	**0.464**	0.464	social_media, analytics_web, web_search, macworld_site, baseball_fantasy, carol_ceo, bartz_ceo, de_la, email_free, marketing_online, obama_barack
Yhoo-tri_biz_df5_sp3_fs	0.533	0.464	
Yhoo-tri_biz_df10_sp3_fs	0.660	0.464	
Yhoo-tri_biz_df20_sp3_fs	0.628	0.464	
Yhoo-bi_df5_sp3_fs	0.511	0.464	mail, google, gmail, twitter, rt, sideline, meme, sports, geocities, bing, music, social, aol, search, news, day, answers, email, open, send, de, web, engine, yslow, bartz, facebook
Yhoo-bi_df10_sp3_fs	0.473	0.464	
Yhoo-bi_df20_sp3_fs	**0.469**	0.464	
Yhoo-bi_df40_sp3_fs	0.545	0.464	
Yhoo-bi_df80_sp3_fs	0.806	0.464	
Yhoo-tri_df10_sp3_fs	**0.486**	0.464	search_rt, flu_swine, search_engine, search_bing, media_social, leapfrog_bing, google_facebook, google_search, bartz_carol, geocities_plug
Yhoo-tri_df20_sp3_fs	0.552	0.464	
Yhoo-tri_df40_sp3_fs	0.627	0.464	
Yhoo-tri_df60_sp3_fs	0.640	0.464	

Table5.3: Experiment results showing models in loose bigram and trigram model with and without topic filtering applied

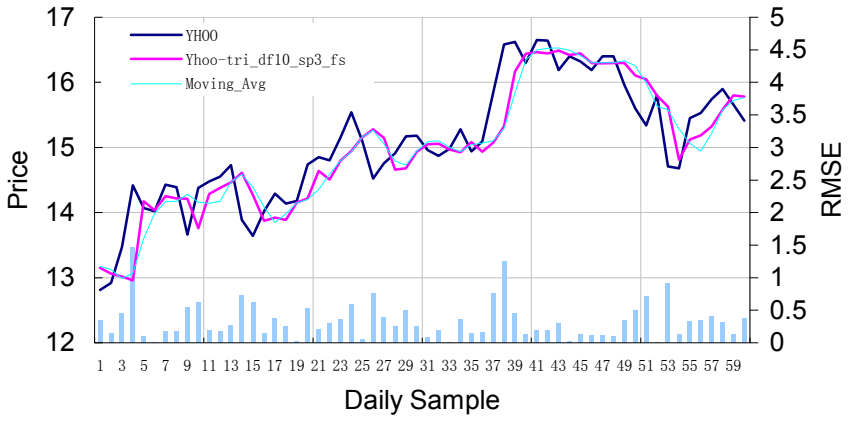


Figure 5.9: Fitted curve from trigram model after wrapper selection, setting “Yhoo-tri_df10_sp3_fs”

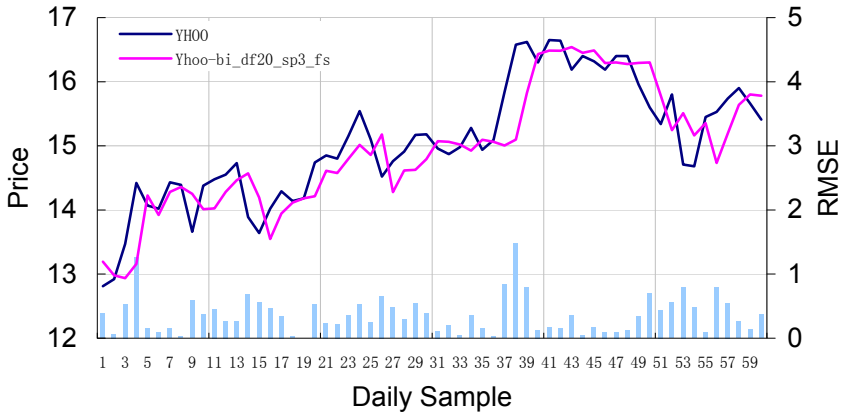


Figure5.10: Fitted curve from bigram model after wrapper selection, setting “Yhoo-bi_df20_sp3_fs”

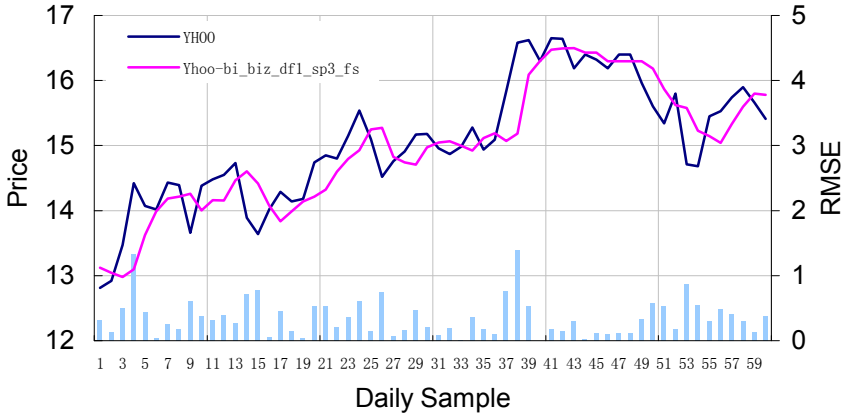


Figure5.11: Fitted curve from trigram model after topic filtering and wrapper selection, setting “Yhoo-bi_biz_df1_sp3_fs”

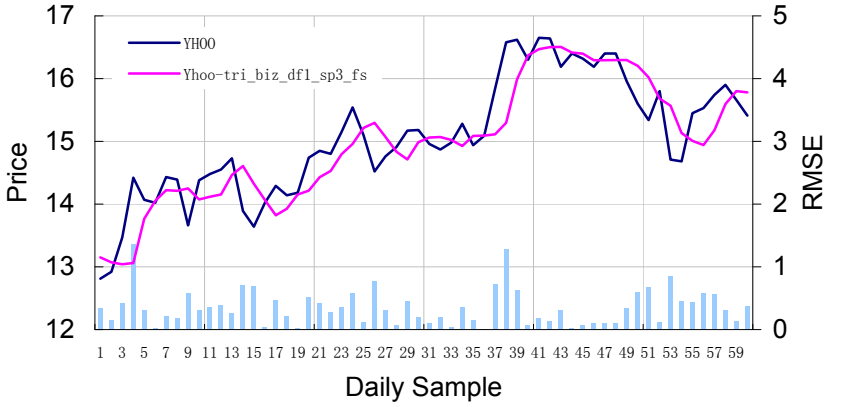


Figure5.12: Fitted curve from bigram model after topic filtering and wrapper selection, setting “Yhoo-bi_biz_df1_sp3_fs”

5.5 Noun Phrase Expansion

As mentioned earlier, we choose Yahoo! Inc. as our main predicting target as it has sufficient coverage in discussion in our dataset but not the most frequent one. Therefore it leaves room to expand the query regarding the company information. In this section we introduce 3 groups of experiments by incorporating more noun phrases into the feature sets. As suggested by Google Finance[1], there are a number of related companies to the Yahoo! stock. In the first group, we expand the noun phrases to include mention of the other two large and highly related companies as suggested in Google Finance: Google Inc. and Microsoft Corporation. We name this group as "*big3*". Further, more companies that are relatively popular in the Internet industry are included: AT&T Inc., eBay Inc., Baidu Inc., Sohu.com Inc. and Apple Inc. We name this group as "*inet*". The third group includes traditional large companies in the IT industry and we name it "*trad*": IBM Corp, Hewlett-Packard Company, Cisco Systems Inc., Oracle Corporation and Intel Corporation. The primary results of the experiments are given in Table 5.4, Table 5.5 and Table 5.6 in which each table corresponds to one expansion group and organised according to the n-gram model and topic filtering.

5.1.1 Simple Noun Counting

We start by studying the simple counting method in the expanded context. As results shown in the results, there is no improvement in RMSE by counting more companies and we have seen significant drops when incorporating the "*inet*" and "*trad*" set of features. As Figure 5.13 suggests, the fitted curve in the "*big3*" model has reasonable trends and can be viewed as smoother comparing to the unexpanded model on several "irregular points" like the ones on day 29 and 38. However, the unbalanced movements in Figure 5.14 and Figure 5.15 show that the simple counting method is very sensitive to the underlying features and their interrelationships. When the features are less related to the target company, they bring more noises so that the models generally fail to capture any true relationship with respect to the stock prices. Although the performance suffers in the current settings, such justification confirms the direct correlations between the companies in the "*big3*" group and more complex relationships to be handled in the other two groups.

---

[1] http://www.google.co.uk/finance?client=ob&q=NASDAQ:YHOO

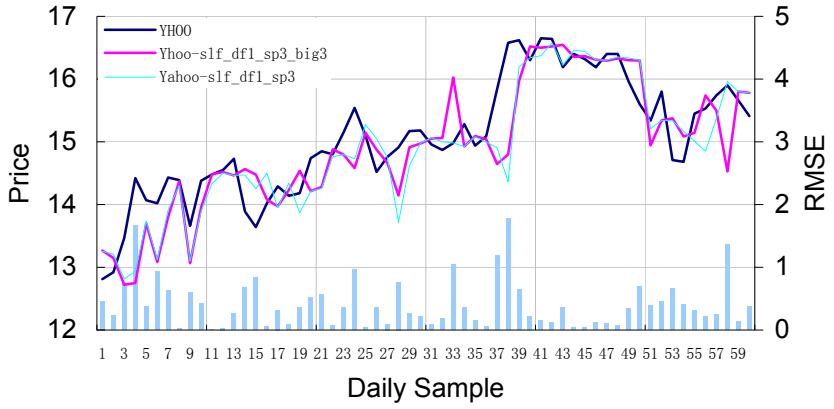


Figure 5.13: Fitted curve from setting “Yhoo-slf_df1_sp5_big3”

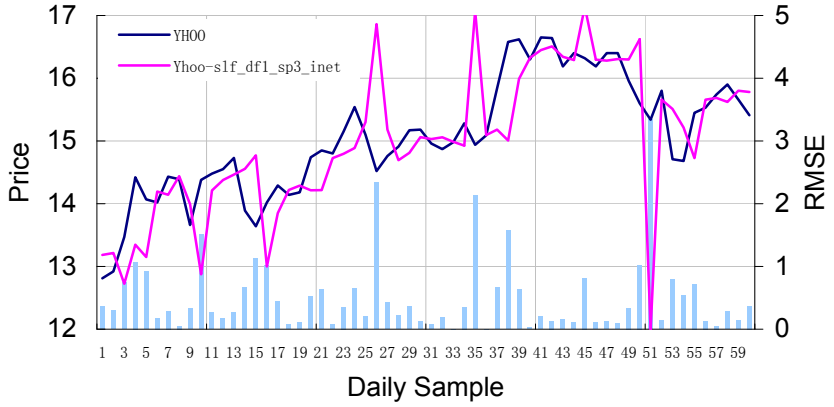


Figure 5.14: Fitted curve from setting “Yhoo-slf_df1_sp5_inet”

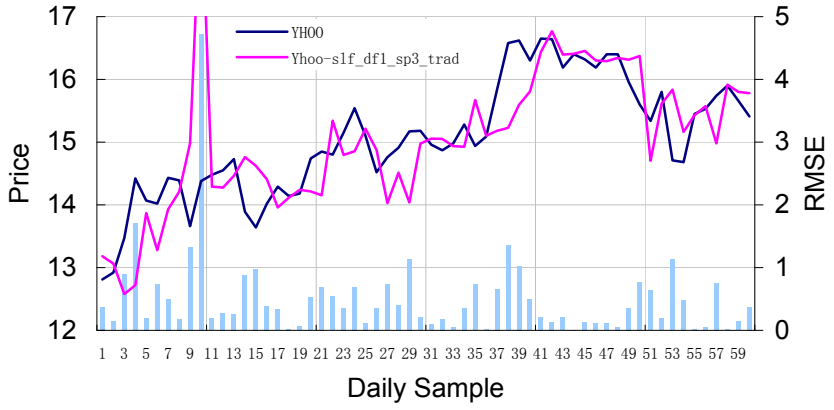


Figure5.15: Fitted curve from setting “Yhoo_slf_df1_sp5_trad”

5.1.2 Loose N-gram Model and Topic Filtering

We further apply the expanded nouns onto the loose n-gram models. The results are generally promising. Models of all three groups built upon the expanded of feature set outperform their original models respectively. The RMSE further come down to the 0.42 to 0.43 level from the 0.44 level of the unexpanded models. From the results one should note that as feature set getting larger the performances of all three groups increase accordingly where the best models come from the largest feature sets. Apparently, this is not the case for the unexpanded models shown in Table 5.2, in which the largest feature sets have decreased performances. This can be viewed as the expansion solely on one company is likely to introduce more noises. In contrast, the regression algorithm actually learns more appropriate relationships from the expanded features from more than one company in which related terms are likely introduced. Such results validate our intention for the noun phrase expansion: the more related company information is incorporated; the better a model would perform. The experiments on the large expanded feature space with improved performance also show the effectiveness of support vector machine to learn complex correlations.

To further evaluate the effectiveness of the noun phrase expansion, in Figure 5.16 to 5.18 we plot the fitted curves, one setting from each group. The best matching curve still come from the bigram model without topic filtering as in Figure 5.16. To be noticed, the curve acts on several turning points without lagging, for instance on day 39, 40, and 41, which are missed out or suffer in lagging by all previous models. These days, which correspond to June 2nd to June 4th, have seen the launch of Microsoft's new search engine Bing, and been discussed thoroughly on the Web. By incorporating the expanded feature set as shown in the top feature section in the result tables, such information is encoded into the model. Therefore, one should note that when the volume of information is sufficient, by incorporating cross-related features, models are able to represent more complex relationships and reflect on price change swiftly. Similarly, we can see that the n-gram features from "*trad*" group are modelled properly comparing to the simple counting methods. Significantly, on day 10 which corresponds to April 10th, there is a large spike on the fitted curve but not on the price curve of Yahoo! Inc. On that day, Oracle Corporation announced an acquisition of Sun Microsystems which greatly impacted the related stock prices and reflected as the top features shown in Table 5.6. However, since the business area of Yahoo! Inc. is less related to those two companies, the model predicted to the wrong direction. Even so, such reaction should be treated as a positive sign since although the target price is not given through training, the model is able to capture the underlying change in data through regression parameter and reflect directly upon prediction. Consequently, we should further understand the importance of relatedness of chosen features to the target company so that the pre-

diction can be “right on the target”.

In summary, the experiments in this section show that incorporate more nouns phrases into the model can impact on the performance. The simple counting method is further shown to be restricted in modelling company information that has complex relationships other than direct correlations. On the other hand, methods using the loose n-gram features with a large frequency threshold are able to capture such relationships through rising number of features and increase in the accuracy and swiftness in responding to changes in direction in the price series. However, we observe that it is important to choose features that do bring valuable information to the target company otherwise the model can predict outside the domain.

Setting	RMSE	Base-line	No. Feature	Top Feagure
Yhoo-slf_df1_sp3_big3	0.575	0.464	4	
Yhoo-bi_biz_df20_sp3_big3	0.564	0.464	213	bing_google, pay_yahoo, iphone_google, seo_yahoo, maps_google, money_microsoft, social_microsoft, jobs_google, networking_yahoo,
Yhoo-bi_biz_df10_sp3_big3	0.560	0.464	1053	
Yhoo-bi_biz_df5_sp3_big3	0.473	0.464	3968	
Yhoo-bi_biz_df1_sp3_big3 (Fig5.18)	**0.457**	0.464	13898	
Yhoo-tri_biz_df20_sp3_big3	0.494	0.464	900	search_term_microsoft, privacy_yahoo_google, loopholes_yahoo_google, bing_yahoo_google, twitter_economy_google, fund_ventures_yahoo
Yhoo-tri_biz_df10_sp3_big3	0.471	0.464	2806	
Yhoo-tri_biz_df5_sp3_big3	**0.445**	0.464	3316	
Yhoo-tri_biz_df1_sp3_big3	0.458	0.464	115995	
Yhoo-bi_df80_sp3_big3	0.660	0.464	358	geocities_yahoo,meme_yahoo, leapfrog_yahoo, eu_microsoft, technet_microsoft, bartz_yahoo, nintendo_microsoft, flickr_yahoo
Yhoo-bi_df40_sp3_big3	0.623	0.464	970	
Yhoo-bi_df20_sp3_big3	0.481	0.464	2812	
Yhoo-bi_df10_sp3_big3	**0.446**	0.464	7938	
Yhoo-tri_df60_sp3_big3	0.474	0.464	2061	twitter_left_google flu_swine_yahoo, engine_microsoft_google, twitter_sideline_yahoo, pirate_cnn_yahoo, bartz_carol, yahoo, bing_search_yahoo
Yhoo-tri_df40_sp3_ big3	0.483	0.464	1920	
Yhoo-tri_df20_sp3_ big3	0.466	0.464	7681	
Yhoo-tri_df5_sp3_big3	**0.434**	0.464	12669	

Table 5.4: Experiment results showing models with expanded features in “*big3*” group.

Setting	RMSE	Baseline	No. Feature	Top Feagure
Yhoo-slf_df1_sp3_inet	0.800	0.464	4	
Yhoo_bi_biz_df40_sp3_inet	0.471	0.464	286	bing_microsoft_google, cash_interest_apple, guide_analytics_google, search_microsoft_google, amazon_sell_google, media_social_apple, guide_analytics_google
Yhoo_bi_biz_df20_sp3_inet	0.469	0.464	192	
Yhoo_bi_biz_df10_sp3_inet	0.466	0.464	962	
Yhoo_bi_biz_df5_sp3_inet	0.481	0.464	3677	
Yhoo_bi_biz_df1_sp3_inet	**0.443**	0.464	17111	

Yhoo_tri_biz_df20_sp3_inet	0.470	0.464	484	web_search, yahoo, plunge_stocks_ebay, motley_stocks_ebay, jobs_ceo_apple,
Yhoo_tri_biz_df10_sp3_inet	0.463	0.464	971	
Yhoo_tri_biz_df1_sp3_inet	**0.436**	0.464	42755	
Yhoo_bi_df80_sp3_inet	0.725	0.464	249	engine_bing_microsoft, rich_apple, bing_ google, links_google, iphone_apple, ipo_ebay, nintendo_microsoft, stumbleupon_ebay, store_apple
Yhoo_bi_df40_sp3_inet	0.465	0.464	497	
Yhoo_bi_df20_sp3_inet	0.493	0.464	1733	
Yhoo_bi_df10_sp3_inet (Fig5.16)	**0.453**	0.464	6838	
Yhoo_tri_df60_sp3_inet	1.358	0.464	276	search_bing_yahoo, twitter_buy_apple, media_social_yahoo, rich_iphone_apple, clean_spring_ebay, rumors_mania_apple,
Yhoo_tri_df40_sp3_inet	0.565	0.464	553	
Yhoo_tri_df20_sp3_inet	0.553	0.464	2241	
Yhoo_tri_df10_sp3_inet	**0.427**	0.464	2241	

Table 5.5: Experiment results showing models with expanded features in "*inet*" group.

Setting	RMSE	Baseline	No. Feature	Top Feagure
Yhoo-slf_df1_sp3_trad	0.843	0.464	5	
Yhoo_bi_biz_df40_sp3_trad	2.255	0.464	77	ibm_oracle, offer_ibm, glad_apple, software_cisco, share_oracle, billion_ibm, software_intel, buying_ibm, social_ibm, technology_oracle, insurance_cisco, woodyswag_ebay
Yhoo_bi_biz_df20_sp3_trad	0.854	0.464	640	
Yhoo_bi_biz_df10_sp3_trad	0.573	0.464	1101	
Yhoo_bi_biz_df5_sp3_trad	**0.433**	0.464	2600	
Yhoo_bi_biz_df1_sp3_trad	0.465	0.464	20489	
Yhoo_tri_biz_df20_sp3_trad	0.509	0.464	431	sun_ibm_oracle, sun_buying_oracle, media_personal_HP,
Yhoo_tri_biz_df10_sp3_trad	0.456	0.464	364	
Yhoo_tri_biz_df5_sp3_trad	0.454	0.464	2143	
Yhoo_tri_biz_df1_sp3_trad	**0.439**	0.464	28438	
Yhoo_bi_df80_sp3_trad	11.140	0.464	38	oracle_ibm, buys_oracle, mysql_oracle, sun_oracle, billion_intel, wind_intel, bought_oracle, microsystems_oracle, dow_cisco, nokia_intel
Yhoo_bi_df40_sp3_trad	1.377	0.464	141	
Yhoo_bi_df20_sp3_trad	1.076	0.464	623	
Yhoo_bi_df10_sp3_trad	**0.559**	0.464	3058	
Yhoo_tri_df60_sp3_trad	1.373	0.464	109	sun_oracle, mysql_oracle, buys_oracle, buy_oracle, rt_oracle, google_afraid_ibm, facebook_afraid_ibm, microsystems_oracle
Yhoo_tri_df40_sp3_trad	1.191	0.464	256	
Yhoo_tri_df20_sp3_trad (Fig5.14)	**0.467**	0.464	1424	

Table5.6 Experiment results showing models with expanded features in "*trad*" group.

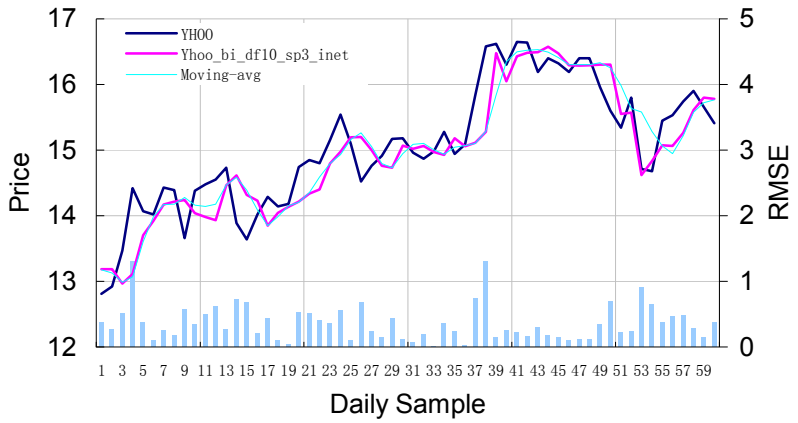


Figure5.16: Fitted curve from setting “Yhoo_bi_df10_sp5_inet”

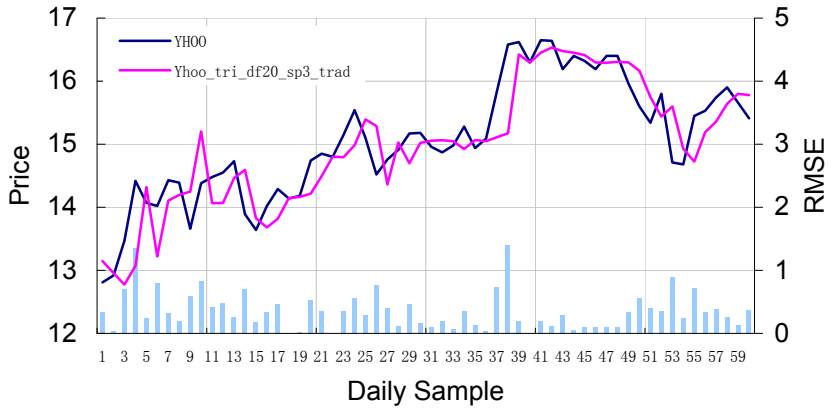


Figure5.17: Fitted curve from setting “Yhoo_tri_df20_sp5_trad”

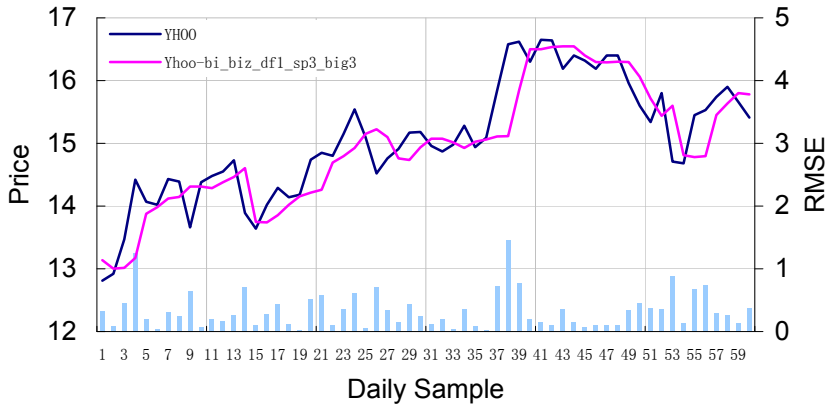


Figure5. 18: Fitted curve from setting “Yhoo_bi_biz_df1_sp5_big3”

5.6 Evaluation on Other Stocks

From the previous experiments, we studied models built towards yahoo prices. For further evaluation, these methods are applied on other stocks and market indices. For individual stocks including Google Inc., Microsoft Corporation and HSBC Holding plc, the same set of methods as Yahoo stock are applied and the ones with best performances are shown in Table 5.7. For stock market indices including NASDAQ Composite, FTSE 100 and S&P 500, we expand the base noun sets to include the names of component companies in these indices and combine the base nouns the same way as before using the loose n-gram model. As the results suggest, all models outperform the baselines except the S&P 500 index. Google Inc. won with a large margin and the others perform slight better than the baselines. As discussed earlier, during experiments we noticed again that Google did have the most mentions in the dataset, in which 145,325 unique bigram features are observed. However, a large number of high frequency features were unrelated to the stock prices like “background color” once occurred more than 1,000 times in 3 days window. This would explain why the best performed method required a topic filtering to remove those noisy terms and left only 1,152 features. Nevertheless, the best model of Google Inc stock is still subject to the noun phrase expansion to the “*inet*” group which was shown to be effective in the previous experiments. When come to comparing the fitted curves, the curves of Google Inc. and Microsoft Corporation have better accuracy in prediction and more synchronised with the true price series whereas the other stocks and indices either have serious lagging issues same as the moving average curves or too many unnecessary fluctuations. This would be largely subject to the fact that these companies including the component companies of the indices have much fewer attentions in the dataset. For example, the term “hsbc” has only 100 mentions per day on average. This makes the information regarding to the companies either incomplete or noisy and thus cannot provide necessary help in the price prediction.

Setting	RMSE	Baseline	No. Feature	Top Feagure
Goog_bi_biz_df1_sp3_inet (Fig5.19)	7.051	8.0438	1152	acquire_google, sell_google, afraid_ibm_microsoft, communication_google, capital_google, fund_google, ceo_google, billion_google
Msft_tri_df40_sp3_big3 (Fig5.20)	0.472	0.4911	1129	e3_microsoft, project_microsoft, geocities_yahoo, jackson_google, encarta_microsoft, controller_microsoft
Hbc_tri_df5_sp3 (Fig5.21)	1.470	1.538	7031	careerist_slash_hsbc, banking_cartasi_hsbc, entrepreneur_young_hsbc, joint_insurance_hsbc, carvalho_budapeste_hsbc
Nasdaq_slf_tri_df1_sp7 (Fig5.22)	32.148	32.321	12804	usd_update_nasdaq, manager_seeking_nasdaq, manager_apple_nasdaq, bank_united_nasdaq, dow_oil_nasdaq
Ftse_tri_df10_sp7 (Fig5.23)	65.152	66.514	5943	life_insurance_aviva, present_smith_tesco, dept_simon_unilever, creme_eggs_cadbury, business_billionaire_bp
Sp500_bi_biz_df5_sp7 (Fig5.24)	16.368	15.878	20436	usb_starbucks, citigroup_dow, revenue_apple, personal_nike, investors_apple, mortgage_gas, profit_citigroup, shares_sun, market_ati

Table 5.7: Experiment results showing models evaluated on other stocks and indices

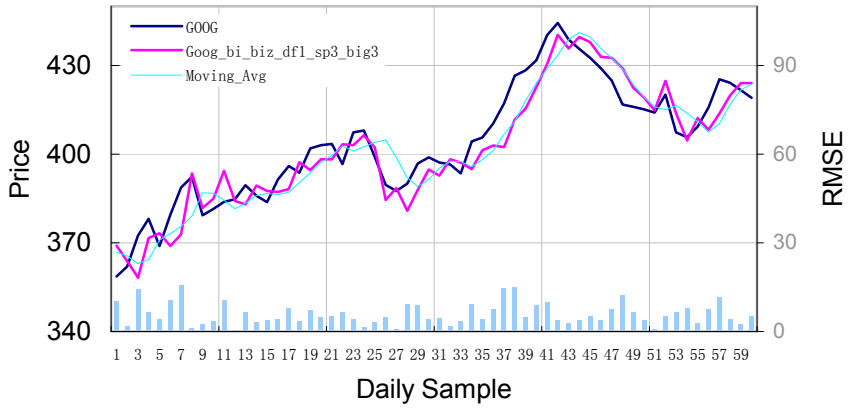


Figure 5.19: Fitted curve from setting “Goog_bi_biz_df1_sp5_big3”

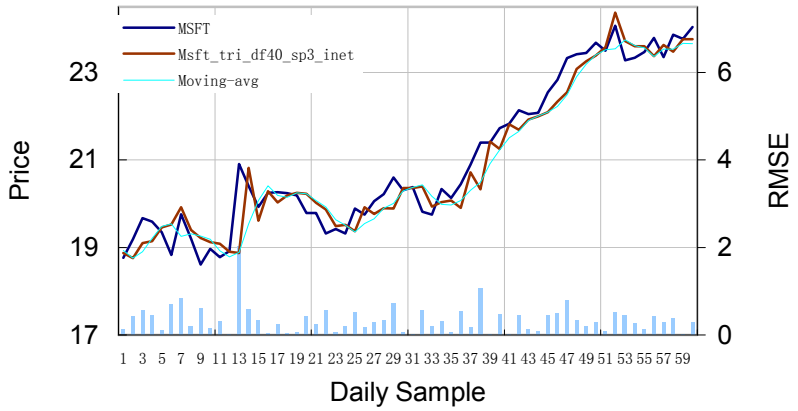


Figure 5.20: Fitted curve from setting “Msft_tri_df40_sp5_inet”

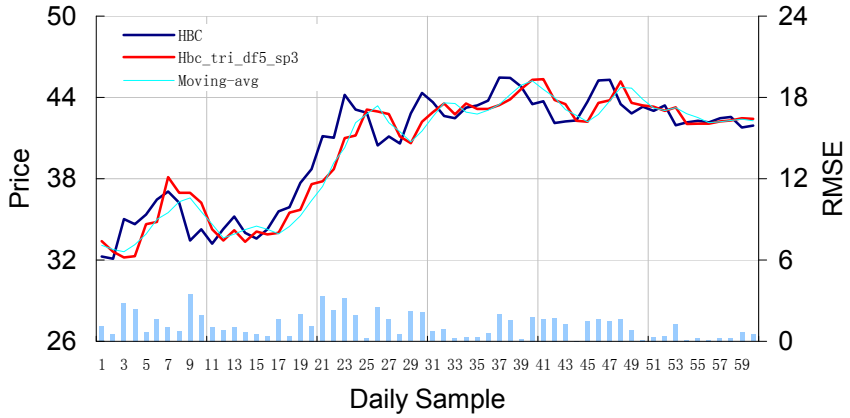


Figure 5.21: Fitted curve from setting “Hbc_tri_df5_sp5”

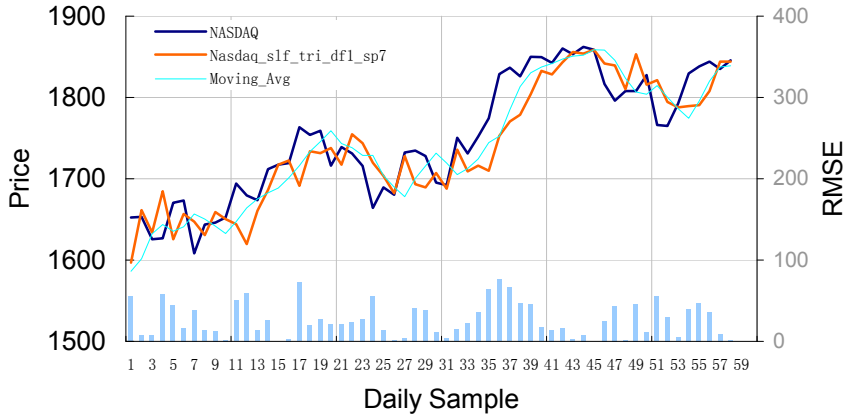


Figure 5.22: Fitted curve from setting “Nasdaq_slf_tri_df1_sp7”

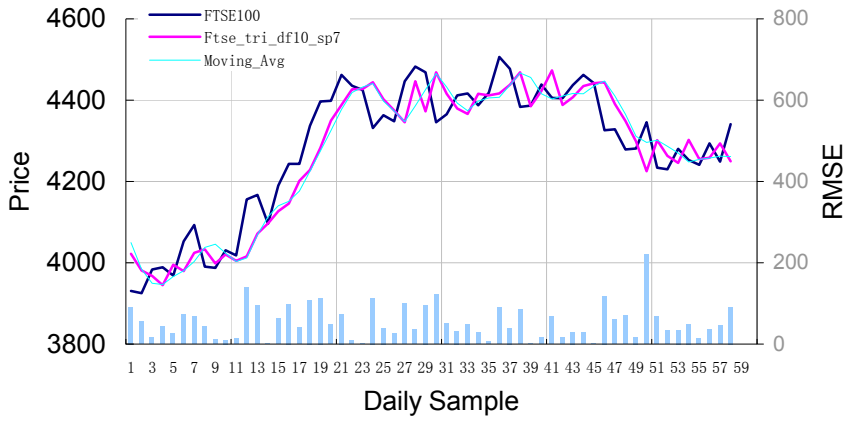


Figure 5.23: Fitted curve from setting “Ftse_tri_df10_sp7”

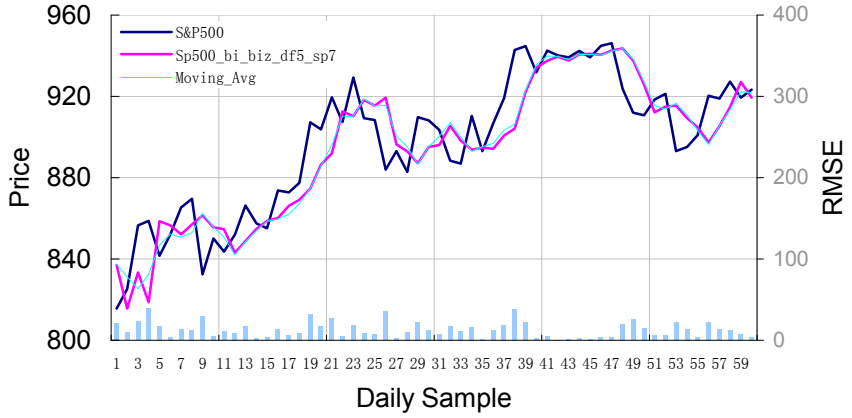


Figure 5.24: Fitted curve from setting “Sp500_bi_biz_df5_sp7”

Chapter 6

Conclusion and Future Work

In this thesis, we attempted the stock market prediction task in the social media context. We utilised a number of different methods as features to represent the textual data crawled from the microblogging service Twitter. We started experiments by using the simple noun counting methods as a direct way of measuring the level of public attention towards the target company and tried to learn the relationships between such measure and the stock prices. The results suggest that such simple method cannot lead to stable performance in prediction. Nevertheless, the promising results of our experiments come from utilising more complex feature representation in which cross-related concepts can be learnt by the model. By utilising a document frequency threshold approach, such methods are able to capture naturally rising concepts and build into the models which lead to improved prediction accuracy. In addition, further experiments on the noun phrase expansion suggests that when more related features are selected from the vast amount of information presented in social media, the predictive model can handle them properly with improved accuracy in prediction.

As this thesis only provides relatively primitive results in order to validate our assumption of predicting stock prices based on public attention presented in social media. Future works can be led to several directions. In our experiments, we only made use of the past 3 days data for training as such data encodes the most recent “memory” of the price movements and is able to maintain future prediction on a reasonable level. However, only utilising such recent history can lose many dramatic price behaviours outside the sampling period. Therefore, future work can investigate into the methods for incorporating longer term history and utilising as prior knowledge. In addition, as our experiments on minor stocks and market indices suggest that the current approach cannot provide enough information of companies that has little public attention. Although this is due to the limitation of current status of social media, mechanism of measuring the sufficiency of information should be devised so that the model can be judged on its effectiveness only on performable areas.

Reference

Abberley, D., Kirby, D., Renals, S. and T. Robinson (1999) The THISL broadcast news retrieval system. In Proc. ESCA ETRW Workshop Accessing Information in Spoken Audio, (Cambridge), pp. 14–19. Section on Query Expansion - concise, mathematical overview.

Anick, P.G., and Vaithyanathan, S. (1997). Exploiting clustering and phrases for context-based information retrieval. Paper presented at the 20th annual international ACM SIGIR conference on research and development, Philadelphia, PA.

Blood, R. (2002), “The Weblog Handbook: Practical Advice on Creating and Maintaining Your Blog”, Cambridge MA: Perseus Publishing.

Boser, E. B., Guyon, I., and Vapnik, V. (1992), “A training algorithm for optimal margin classiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory”, pages 144-152. ACM Press.

Chiang, W.-C., Urban, T., and Baldridge, G. (1996). A neural network fund net asset approach to mutual value forecasting. Omega.

Drucker, H, Burges, C. J. C., Kaufman, L., Smola, A. and Vapnik, V. (1997), “Support Vector Regression Machines”, Advances in Neural Information Processing Systems 9, NIPS, 155-161, MIT Press.

Dumais, S. and Chen, H. (2000), Hierarchical classification of Web content. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval.

Fama, E. (1965), "The Behavior of Stock Market Prices". *Journal of Business* **38**: 34–105.

Fung PC., Yu GX., Lam JW. (2003), Stock prediction: Integrating text mining approach using real-time news, Proceedings of IEEE International Conference on Computational Intelligence for Financial Engineering.

Fung, G. P. C., Yu, J. X. and Lam, W. (2002) “News sensitive stock trend prediction,” in Proceedings of the 6th Pacific-Asia Conference on Knowledge. Discovery and Data Mining, Taipei, Taiwan, pp. 289–296.

Fung, GX Yu, JW Lam (2003) “Stock prediction: Integrating text mining approach using real-time news,” in Proceedings of the 7th IEEE International Conference on Computational Intelligence for Financial Engineering, Hong Kong, China, pp. 395–402.

Gutlein M., Frank Eibe, Hall Mark, Karwath Andreas (2009), “Large scale attribute selection using wrappers, In Proc. of the IEEE Symposium on Computational Intelligence and Data Mining”, *Mining and Knowledge Discovery*, **2(2)**:121–167.

Herring, C. S., Scheidt, A. L. & Bonus, S. (2004), “Bridging the Gap: A Genre Analysis of Weblogs”, In Proceedings of the 37th Hawaii International Conference on System Sciences.

Herring, C. S.,Scheidt, A.L. & Bonus, S. (2004) ‘Bridging the Gap: A Genre Analysis of Weblogs’, In Proceedings of the 37th Hawaii International Conference on System Sciences.

Kovach, B. and Rosenstiel. T. (1999) Warp Speed: America in the Age of Mixed Media. Century Foundation Press,

Lavrenko, V., Schmill, M. D., Lawire, D., Ogivie, P. and Jensen, D. and Allan, J. (2000) “Mining of concurrent text and time series,” in Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining Workshop on Text Mining, Boston, MA, USA, , pp. 37–44.

Lawrence, S., Giles, C. L., and Tsoi, A.-C. (1997). Lessons in neural network training: Overfittting may be harder than expected. In Proceedings of the fourteenth national conference on artificial intelligence, AAAl-97 (pp. 540–545).

Lee, C (2009), “Using support vector machine with a hybrid feature selection method to the stock trend prediction. Expert Systems with Applications: An International Journal. Volume 36 , Issue 8. pp. 10896-10904.

Leskovec, J., Backstrom, L., and Kleinberg J.(2009), “Meme-tracking and the dynamics of the news cycle”, Proc. 15th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining.

Li, Q. and Li, J (2005), An Efficient Topic-Specific Web Text Filtering Framework, APWeb 2005: 157-163

Malkiel, B. G. (1973), “A Random Walk Down Wall Street (6th ed.)”, W.W. Norton & Company, Inc.. ISBN 0393062457.

Min, J. H., and Lee, Y.-C. (2005). Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. Expert Systems with Applications, 28(4), 603–614.

Mishne G. & Rijke de M. (2006), “A Study of Blog Search”, In Proceedings of 28th European Conference on Information Retrieval (ECIR).

Mishne, G. (2006), “Information Access Challenges in the Blogspace”, In Proceedings of International Workshop on Intelligen Information Access 2006.

Samuelson Paul (1965), “Proof That Properly Anticipated Prices Fluctuate Randomly” *Industrial Management Review* **6**: 41–49.

Shen, D., Sun, J.-T., Yang, Q., and Chen, Z. (2006), “Latent Friend Mining From Blog Data”, in ICDM'06: Proceedings of the Sixth International Conference on Data Mining, pp. 552–561, Washington DC, IEEE Computer Society.

Smola, AJ and Schölkopf, B. (2004) A tutorial on support vector regression. Statistics and Computing.

Smola, AJ and Schölkopf, B. (2004), A tutorial on support vector regression, Statistics and Computing, Springer

Tolle, K.M. and Chen, H. (2000), Comparing Noun Phrasing Techniques for Use with Medical Digital Library Tools. Journal of the American Society for Information Science.

Vapnik, V. (1995) The Nature of Statistical Learning Theory. Springer-Verlag, New York, NY.

Yang, Y., Pedersen J.P. (1997) A Comparative Study on Feature Selection in Text Categorization Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), pp412-420.

Yiming Yang , Xin Liu, (1999) A re-examination of text categorization methods, Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp.42-49, August 15-19, Berkeley, California, United States.

Zhang X. and Zhu, X. (2007) A New Type of Feature - Loose N-Gram Feature in Text Categorization, Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis, Part I.