The Predicting Power of Textual Information on Financial Markets

Gabriel Pui Cheong Fung†, Jeffrey Xu Yu†, Hongjun Lu‡

***Abstract*— Mining textual documents and time series concurrently, such as predicting the movements of stock prices based on the contents of the news stories, is an emerging topic in data mining community. Previous researches have shown that there is a strong relationship between the time when the news stories are released and the time when the stock prices fluctuate. In this paper, we propose a systematic framework for predicting the tertiary movements of stock prices by analyzing the impacts of the news stories on the stocks. To be more specific, we investigate the immediate impacts of news stories on the stocks based on the Efficient Markets Hypothesis. Several data mining and text mining techniques are used in a novel way. Extensive experiments using real-life data are conducted, and encouraging results are obtained.**

I. Introduction

In the financial markets, the movements of the prices are the consequences of the actions taken by the investors on how they perceive the events surrounding them as well as the financial markets. Investors' decisions on bidding, asking or holding the securities are greatly influenced by what others said and did within the financial markets [1], [2]. The emotions of fear, greed, coupled with subjective perceptions and evaluations of the economic conditions and their own psychological predispositions and personalities, are the major elements that affect the financial markets' behaviors [3], [4].

Yet, human behaviors are not random. People's actions in the financial markets, although occasionally irrational, are predominantly understandable and rational with respect to the social structure, social organization, perceptions and collective beliefs of this complex arena [1], [2], [5], [6]. At times, collective movements are launched which are in turn based on a group beliefs about how the markets will act or react [3], [4]. In these instances, trends develop which can be recognized, identified and anticipated to continue for some periods.

Nowadays, an increasing amount of crucial and valuable information highly related to the financial markets is widely available on the Internet.[1] However, most of these information are in textual format, such as news stories, companies' reports and experts' recommendations. Hence, extracting valuable information and figuring out the relationship between the extracted information and the financial markets are neither trivial nor simple.

†Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, {pcfung, yu}@se.cuhk.edu.hk

‡Department of Computer Science, The Hong Kong University of Science and Technology, luhj@cs.ust.hk

[1]E.g. Wall Street Journal: www.wsj.com, Financial Times: www.ft.com, Reuters: www.reuters.com, Bloomberg: www.bloomberg.com, CNN: www.cnnfn.com, etc

In this paper, we investigate how to utilize the rich textual information in predicting the financial markets. In contrast to the traditional time series analysis, where predictions are made based solely on the technical data (historical movements of the time series) and/or the fundamental data (the companies' profiles), this paper focuses on the problem of predicting the impacts of the textual information on the financial markets. To be more specific, real-time news stories and intra-day stock prices are used to denote respectively the information obtained from textual documents and the the movements of the financial markets. In other words, predictions are made according to the contents of news stories, rather than using the numerical data. This kind of problem is sometimes known as mining time series and textual documents concurrently [7], [8], [9], which is an emerging topic in the data mining community nowadays [10], [11], [12], [13]. The main contributions of this paper are summarized below:

1) **Figuring out the tertiary movements of the stock prices** A tertiary movement lasts for less than three weeks, which denotes the short-term stock market behavior [14], [15], [16]. In other words, tertiary movement is highly affected by the events surrounding the financial market. A new piecewise linear approximation algorithm, called *t-test based split and merge segmentation algorithm*, is proposed for figuring out the tertiary movements automatically.
2) **Detecting the relationship between the events mentioned in the news stories and the tertiary movements of stock prices** We have to correctly *align* and *select* the news stories to the tertiary movements of the stock price such that the aligned news stories are most likely to trigger or support the movements of the trends. The alignment process is based on the *Efficient Market Hypothesis* [1], [2] and the selection of the useful news stories is based on a χ^2 *estimation on the keywords distribution over the entire document collection.*
3) **Predicting the impact of a newly released news story on the stock price** Three kinds of impact are defined: `positive`, `negative` and `natural`. A piece of news story is said to have `positive` (or `negative`) impact if the stock price rises (or drops) significantly for a period after the news story is released; otherwise, if the the stock price does not fluctuate even after the news story is released, we said that the impact of the news story is `natural`. The major learning and prediction process is based on the text classification algorithm, *Support Vector Machines* [17].

The rest of this paper is organized as follows. Section II reviews the major preliminaries related to the discussion of this paper. Section III presents our proposed system. Section IV evaluates various aspects related to our proposed approach. A summary and conclusion is given in Section VI.

II. Preliminaries

The first systematic examination against the impacts of textual information on the financial markets is conducted by Klein and Prestbo [18]. Their survey consists primarily of a comparison of the movements of Dow Jones Industrial Average[2] with general news during the period from 1966 to 1972. The news stories that they have taken into consideration are the stories appearing in the "What's New" section in the *Wall Street Journal*, as well as three featured stories[3] carried on the Journal's front page. The major criticism of their study is that too few news stories are taken into consideration in each day. It is rather simple to assume that stories carried on the front page of the Wall Street Journal are enough for summarizing and reflecting the information appear in the whole newspaper. Interestingly, even with such a simple setting, Klein and Prestbo found that the pattern of directional correspondence, whether upwards or downwards, between the flow of the news stories and stock price movements manifested itself 80% of the time. Their findings strongly suggest that news stories and financial markets tend to move together.

Fawcett and Provost [10] formulate an *activity monitoring task* for predicting the stock price movements based on the content of the news stories. Activity monitor task is defined as the problem that involves monitoring the behaviors of a large population of entities for interesting events which require actions. The objective of the activity monitoring task is to issue alarms accurately and quickly. In the stock price movements detection, news stories and stock prices for approximately 6,000 companies over three months period are archived. An interesting event is defined to be a 10% change in stock price which can be triggered by the content of the news stories. The goal is to minimize the number of false alarms and to maximum the number of correctly predicted price spikes. It is worth noting that, the authors only provide a framework for formulating this predicting problem. The implementation details and an in-depth analysis are both missing. Perhaps this is because their main focus is not on examining the possibility of detecting stock price movements based on news stories, but is on outlining a general framework for formulating and evaluating the problems which require continuous monitoring their performance.

Thomas and Sycara [12] predict the stock prices by integrating the textual information that are downloaded from the web bulletin boards[4] into trading rules. The trading rules are derived by genetic algorithms based on numerical data. For the textual data, a maximum entropy text classification approach [19] is used for classifying the impacts[5] of the posted messages on the stock prices. For the trading rules, they are constructed by genetic algorithms based on the trading volumes of the stocks concerned, as well as the number of messages and words posted on the web bulletin boards per day. A simple market simulation is conducted. The authors reported that the profits obtained increased up to 30% by integrating the two approaches rather than using either of them. However, no analysis on their results is given.

Wuthrich et al. [13] develop an online system for predicting the opening prices of five stock indices[6] by analyzing the contents of the electronic stories downloaded from the Wall Street Journal. The analysis is done as follows: for each story, keywords are extracted and weights are assigned to them according to their significance in the corresponding piece of news story and on the corresponding day. By combining the weights of the keywords and the historical closing prices of a particular index, some probabilistic rules are generated using the approach proposed by Wuthrich [20], [21]. Based on these probabilistic rules, predictions on at least 0.5% price changes are made. The weaknesses of their system is that only the opening prices of financial markets could be predicted. Some others more challenging and interesting issues, such as intra-day stock price predictions, could not be achieved.

Following the techniques proposed by Wuthrick et al., Permunetilleke and Wong [11] repeat the work but with different a domain. News headlines (instead of news contents) are used to forecast the intra-day currency exchange rate (instead of the opening prices of stock indices). These news headlines belong to world financial markets, political or general economic news. They show that on a publicly available commercial data set, the system produces results are significantly better than random prediction.

Lavrenko et al. [9] propose a system for predicting the intra-day stock price movements by analyzing the contents of the real-time news stories. Analyst is developed based on a language modeling approach proposed by Ponte and Croft [22]. While a detailed architecture and a fruitful discussion are both presented in their paper, the following questions are unanswered: The authors claim that there should be a period, t, to denote the time for the market to absorb any new information (news stories) release, where t is defined as five hours. We have to admit that the market may spent time to digest information. However, such a long period may contradicts with most economic theories [1], [2]. In addition, news stories may frequently classify to trigger both the rise and drop movements of the stock prices in the training stage, which is a dilemma. Finally, in their evaluations, the impact of the news stories are "immediate" (without 5 hours time lag). This contradicts to the training phase of the system.

III. The Proposed System

In this paper, we are interested in determining whether a news story would have any impacts on the stock prices, and

[2] A financial index which composed of 30 blue-chip stocks listed on the New York Stock Exchange.

[3] Klein and Prestbo did not describe in details how they selected these three stories among all stories carried on the Journal's front page.

[4] Thomas and Sycara chose Forty discussion boards from `www.ragingbull.com`

[5] Two impacts are defined in their paper: up and down.

[6] These five stock indices are: Dow Jones Industrial Average, Nikkei 225, Financial Times 100 Index, Hang Seng Index and Singapore Straits Index

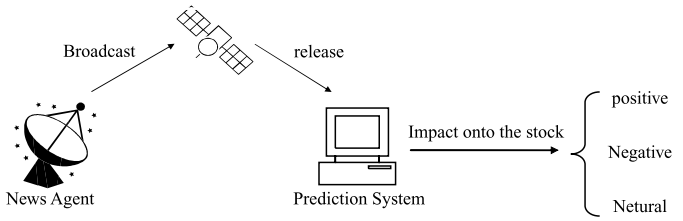


Fig. 1. The operation of the proposed system. After we received a news story from a news agent, we determine which of the three impact it has: `positive`, `negative` or `neutral`. A news story is said to have `positive impact` (or `negative impact`) if the stock price rise (or drop) signifi cantly for a period after the news story is released. If the stock price does not change after the news story is released, then the news story is regarded as `neutral`.

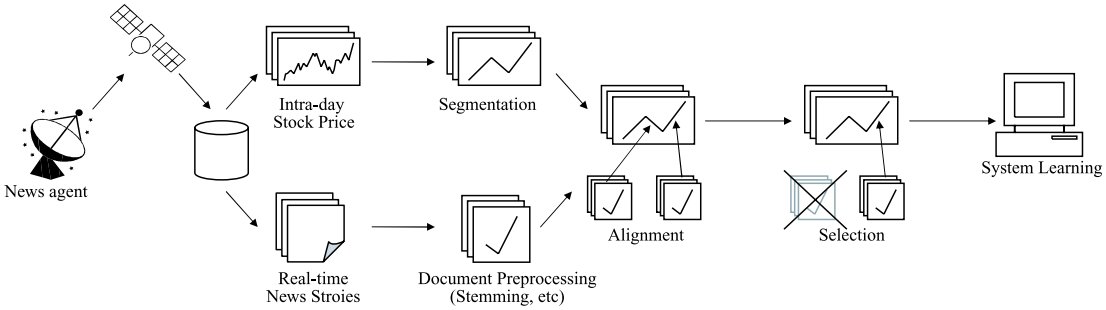


Fig. 2. The architecture of the proposed system. Four major processes are defi ned: 1) News Stories Alignment; 2) Time Series Segmentation; 3) Time Series Segmentation and 4) System Learning.

if so, what kinds of *impact* is this news story. Three impacts are defined: `positive`, `negative` and `neutral`. A news story is said to have `positive impact` (or `negative impact`) if the stock price rise (or drop) significantly for a period, T, after the news story has been broadcasted. If the stock price does not change after the news story is broadcasted, then the news story is regarded as `neutral`. Figure 1 illustrates the motivation described here.

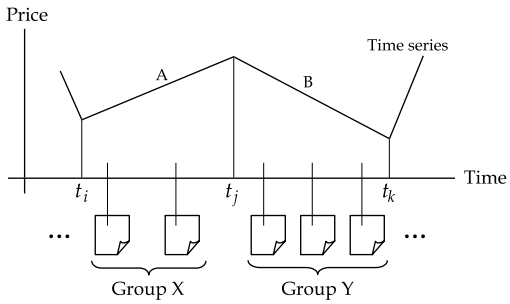
Figure 2 shows the architecture of the proposed system. For any prediction system to operate successfully, we first archive and label some sets of data and present them to the system for learning their relationships. These data are known as training data. The training data that we have taken are real-time news stories and intra-day stock prices. Since there are too many news stories and stocks in the market, such that it is impossible for us to read through the news stories one by one and classify their impact manually, we therefore must have a heuristics for selecting them automatically. We explain the details of the system in the following sections.

A. News Stories Alignment

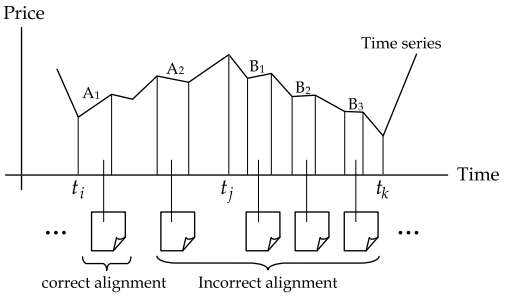
In order to obtain a set of reliable training data, we have to correctly align the news stories to the stock trend such that the aligned news stories is believed to trigger or support the movements of the trends. For aligning news stories to the stock time series, there could be three different formulation under different assumptions. They are further explained below:

1) **Formulation 1 – Observable Time Lag** In this formulation, there is a time lag between the news story is broadcasted and the stock price moves. It assumes that the stock market needs a long time for absorbing the new information. Let us take Figure 3 (a) to illustrate this idea. In this formulation, the Group X (news stories), is responsible for triggering Trend B, while Group Y does nothing with the two trends. Some reported works used this representation [9], [11], [12].
2) **Formulation 2 – Effi cient Market** In this formulation, the stock price moves as soon as after the new story is released. No time lag is observed. This formulation assumes that the market is efficient and no arbitrary opportunity normally exists. To illustrate this idea, let us refer to Figure 3 (a). Under this formulation, Group X is responsible for triggering Trend A, whild Group Y is responsible for triggering Trend B.
3) **Formulation 3 – Reporting** In this formulation, new stories are released only after the stock price has moved. This formulation assumes that the stock price movements are neither affected nor determined by any new information. The information (e.g. news stories) are only useful for *reporting* the situation but not *predicting* the future. Again, let us use Figure 3 (a) to illustrate this idea. Under this formulation, Group Y is responsible for accounting why Trend A would happened. Group X does nothing with the two trends.

Different scholars may in favor of one of the formulation. It is difficult, if not impossible, for finding a completely consensus. In this paper, we take the second formulation (Formulation 2 – Efficient Market), which is based on the Efficient Market Hypothesis. Thanks to Efficient Market Hypothesis, which states that the current market is an efficient information processor, such that it reflects the assimilation of



(a) An ideal situation.



(b) The truth situation.

Fig. 3. The alignment process. On the left: for the ideal situation, stories broadcast under the time series should be responsible for the fluctuation of the time series in that period. On the right: in reality, stock time series exhibit a high level of noise such that the impact of most stories are determined incorrectly.

all of the information available *immediately* [23], [24], we therefore align the news stories to the time series using the second formulation.

More formally, let d_i be a news story; $\mathcal{D}$ denote all of the news stories archived; $\mathcal{D}_{S_k}$ denote the documents that are aligned to segment S_k; $t_{rel}(d_i)$ denote the timestamp when the document d_i is released; $t_{begin}(S_k)$ and $t_{end}(S_k)$ denote the timestamp of segment S_k begin and the timestamp of segment S_k end, respectively. According to the second formulation that that documents which are broadcasted within a segment are aligned back to that segment:

$$d_i \in \{\mathcal{D}_{S_k} \mid t_{rel}(d_i) \geq t_{begin}(S_k) \text{ and } t_{rel}(d_i) < t_{end}(S_k)\} \quad (1)$$

However, no matter which formulation we take, note that all stock time series contain a high level of noise. Since every stock time series contains a high level of noise, such that even though the general trend is rising (or dropping), some dropping (or rising) segments can be observed. If we simply align the news stories based on the type of the time series segments (rise or drop), wrong alignment must be resulted. Figure 3 (b) illustrates this idea. In Figure 3 (b), even though the general trends from t_i to t_j is rising, the segment A_2 is slightly dropping. The news story releases under segment A_2 should be regarded as having `positive impact` (general trend) rather than having `negative impact` (exact observation). Similar situation is observed from t_j to t_k.

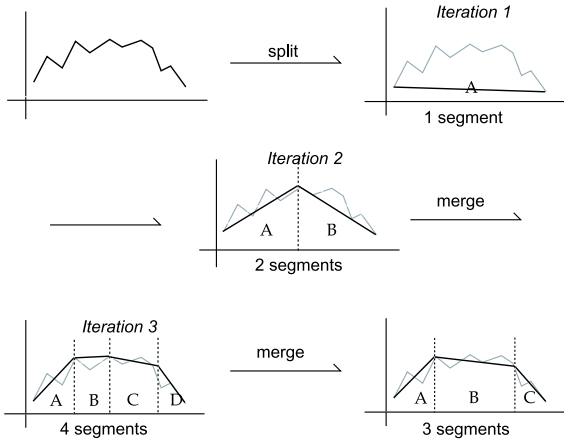


Fig. 4. General idea of the t-test based split and merge segmentation algorithm. The splitting phase aims at discovering all of the possible trends on the time series, while the merging phase aims at avoiding over-segmentation.

Algorithm 1 segment(T) – segment a time series T

1: T_{tmp} = split($T[t_0, t_n]$);
2: T_{final} = merge(T_{tmp});
3: **return** T_{final}

In order to remedy the above phenomenon, a higher level re-describing the time series into trends is necessary, e.g. re-describing Figure 3 (b) to Figure 3 (a) is necessary. This process is also known as time series segmentation. We provide our segmentation algorithm in the next section.

B. Time Series Segmentation

As with most data mining problems, data representation is one of the major elements to reach an efficient and effective solution. Since all stock time series contains a high level of noise, a high level time series segmentation is necessary for recognizing the trends on the times series. A sound time series representation involves issues such as recognizing the significant movements or detecting any abnormal behaviors, so as to study and understand its underlying structure.

Piecewise linear segmentation, or sometimes called piecewise linear approximation, is one of the most widely used technique for time series segmentation, especially for the financial time series [9], [25], [26]. It refers to the idea of representing a time series of length n using K straight lines, where $K \ll n$ [27], [28]. Most studies in this area are pioneered by Pavlidis et al. [28] as well as Dedua and Harts [29].

In this paper, we propose a t-test based split-and-merge piecewise linear approximation algorithm. The splitting phase aims at discovering trends on the time series, while the merging phase aims at avoiding over-segmentation. Figure 4 and Algorithm 1 illustrate the general idea of the proposed segmentation algorithm.4

t-test Based Split and Merge Segmentation Algorithm – Splitting phase

Let $T = \{(t_0, p_0), (t_1, p_1), \ldots, (t_n, p_n)\}$ be a financial time series of length n, where p_i is the price at time t_i for $i \in [0, n]$.

Initially, the whole time series is regarded as a single large segment, and is represented by a straight line joining the first and the last data points of the time series (Figure 4). In order to decide whether this straight line (segment) can represent the general trend of the time series, a one tail *t*-test is formulated:

$$\begin{aligned} H_0 &: \varepsilon = 0 \\ H_1 &: \varepsilon > 0 \end{aligned} \tag{2}$$

where ε is the expected mean square error of the straight line with respect to the actual fluctuation on the time series:

$$\varepsilon = \frac{1}{k} \cdot \sum_{i=0}^{k} (p_i - \hat{p}_i)^2 \tag{3}$$

where k is the total number of data points within the segment, $\hat{p}_i$ is the projected price of p_i at time t_i. The required t-statistics is:

$$t = \frac{\varepsilon}{\sqrt{\hat{\sigma}^2/n}} \tag{4}$$

where $\hat{\sigma}$ is the standard deviation of the mean square error, ε. The t-statistics is therefore compared with the t-distribution with $n-1$ degree of freedom using $\alpha = 0.05$. In other words, there is a probability of 0.05 that the null hypothesis ($H_0 : \varepsilon = 0$ in Equation (2), would be accepted given that it is incorrect.

The motivation of this formulation is that if the null hypothesis ($H_0 : \varepsilon = 0$) in Equation (2) is accepted, then the mean square error between the actual data points and the projected data points should be very small. Thus, the straight line, which is formulated by joining the first and the last data points of the segment, should be well enough to represent the trends of the data points in that segment. In contrast, if the alternative hypothesis is accepted ($H_1 : \varepsilon > 0$), then a single straight line is not well enough to represent the trend of the data points in the corresponding segment.

Let us consider for the case where the null hypothesis is rejected. If the null hypothesis is rejected, then the straight line is split at the point where the error norm is maximum, i.e. $\max_i \{(p_i - \hat{p}_i)^2\}$, and the whole process will be executed recursively on each segment (Figure 4 (b) – (c)). Algorithm 2 outlines the procedure of the splitting phase.

Algorithm 2 split($T[t_a, t_b]$) – split a time series T of length n from time t_a to time t_b where $0 \leq a < b \leq n$

```
T_temp = ∅
ε_min = ∞;
ε_total = 0;
for i = a to b do
    ε_i = (p_i − p̂_i)^2;
    if ε_min > ε_i then
        ε_min = ε_i;
        t_k = t_i;
    end if
    ε_total = ε_total + ε_i;
end for
ε = ε_total/(t_b − t_a);
if t-test.reject(ε) then
    T_temp = T_temp ∪ split(T[t_a, t_k]);
    T_temp = T_temp ∪ split(T[t_k, t_b]);
end if
return T_temp;
```

Algorithm 3 merge(T) – attempt to merge two adjacent segments on the time series T

```
while true do
    ε_min = ∞;
    repeat
        i = 0
        ε_i = Σ_{j=t'_i}^{t'_{i+2}} (p_j − p̂_j)^2;
        if ε_min > ε_i then
            ε_min = ε_i;
            k = i + 1;
        end if
    until end of the time series
    if t-test.accept(ε_min) then
        drop (t_k, p_k);
    else
        break;
    end if
end while
return T
```

t-test Based Split and Merge Segmentation Algorithm – Merging phase

After the splitting phase, *over-segmentation* will frequently occur. Over-segmentation refers to the situation where there exist two adjacent segments such that their slopes are similar, and they should be merged to from a single large segment. Let us refer to Figure 4 again. If we only perform the splitting phase, four segments would be resulted. However, note that the slopes of segment A_2 and segment B_1 are very similar. Hence, merging them is possible. After merging A_2 and B_1, three segments are remained. All of the segments now have different slopes. In other words, merging phase aims at combining all of the adjacent segments, provided that the mean square error, ε, would still be accepted by the t-test after merging. The hypothesis for the t-test is the same as Equation (2).

More formally, consider the time series T which has been transformed into another time series $T_{temp} = \{(t'_0, p'_0), (t'_1, p'_1), \ldots, (t'_m, p'_m)\}$ of length m after the splitting phase, such that $m \ll n$. Define $S_i = \{(t'_i, p'_i), (t'_{i+1}, p'_{i+1})\}$ as a segment in T_{temp}. If the null hypothesis over two adjacent segments, S_i and S_{i+1}, is accepted, then these two segments are regarded as a *candidate merging pair*. Let $\mathcal{L}_{merge}$ be a list containing all of these candidate merging pairs. One of the candidate merging pair resides in $\mathcal{L}_{merge}$ would be selected to merge if merging of it would be resulted in the minimum increase in the total error norm. The whole process is executed continuously until the t-test over all of the segments on the time series is rejected, i.e. $\mathcal{L}_{merge} = \phi$. Algorithm 3 illustrates

TABLE I
A 2×2 CONTINGENCY TABLE SUMMARIZED THE DISTRIBUTION OF FEATURE f_j IN THE DOCUMENT COLLECTION. THIS TABLE COULD BE MODELED BY A χ^2 DISTRIBUTION WITH ONE DEGREE OF FREEDOM.

	#documents have f_j	#documents do not have f_j
Segment $= S_k$	case 1	case 3
Segment $\neq S_k$	case 2	case 4

the whole procedure of merging phase.

C. Useful News Stories Selection

In reality, many news stories are valueless in prediction, i.e. they do not contribute to the prediction of the stock prices. In this section, we present how to select the valuable news stories.

Define $features$ to be any words in the news story collection. Let f_j be a feature in the news story collection. Recall that news story that are released within a segment are aligned back to that segment, i.e. $d_i \in \{\mathcal{D}_{S_k} \mid t_{rel}(d_i) \geq t_{begin}(S_k)$ and $t_{rel}(d_i) < t_{end}(S_k)\}$. By counting the presence or absence of f_j appearing during a given segment, a statistic model for discrete events could be formulated. In such model, the frequency of any feature appearing within the news story collection would be random with unknown distribution. In a model that features are emitted at a random process, two assumptions could be made: 1) The process of generating the features is *stationary*; and 2) The occurrence of every feature is *independent* of each other, i.e. $P(f_a) = P(f_a|f_b)$.

For the first assumption, if a feature is stationary, then in any arbitrary period, the probability of getting it is the same as at any other periods. In other words, if the probability of a feature appearing in some periods change dramatically, we can conclude that this feature exhibit an abnormal behavior in those periods, and it would be regarded as an important feature in there. Specifically, by counting the number of documents that: 1) contains feature f_j and is in Segment S_k; 2) contains feature f_j but is not in Segment S_k; 3) does not contain feature f_j but is in segment S_k; and 4) does not contain feature f_j and is not in segment S_k, a 2×2 contingency table could be formulated (Table I). Note that this table could be modeled by a χ^2 distribution with one degree of freedom.

For the second assumption, it is known as the independent assumption of feature distribution, which is a common assumption in text information management, especially for information retrieval, clustering and classification. Researches show that this assumption will not harm the system performance [30], [31], [32], [33]. Indeed, maintaining the dependency of features is not only extremely difficult, but also may easily degrade the system performance [34], [32], [33], [35].

For each feature f_j under each segment S_k, we calculate its χ^2 value, i.e. $\chi^2(f_j, S_k)$. If it is above a threshold, α, i.e. $\chi^2(f_j, S_k) \geq \alpha$, we conclude that the occurrence of feature f_j in segment S_k is significant, and this feature is appended into a feature list, $\mathcal{L}_{feature,S_k}$. $\mathcal{L}_{feature,S_k}$ stores all the features in which their occurrence in segment S_k are significant:

$$f_j \in \mathcal{L}_{feature,S_k} \text{ if } \chi^2(f_j, S_k) \geq \alpha \quad (5)$$

Define $\mathcal{D}_R$ and $\mathcal{D}_D$ be two sets containing the documents that support the rise movement and drop movement, respectively. Hence, these two sets are served as the positive training examples for the rise and drop trends. A document, d_i, which belongs to segment S_k ($d_i \in \mathcal{D}_{S_k}$), would be assigned to $\mathcal{D}_R$ if and only if the slope of S_k is positive and d_i contains a feature listed in $\mathcal{L}_{feature,S_k}$ (i.e. $d_i \in \mathcal{D}_R$ iff $f_j \in \{\mathcal{L}_{feature,S_k}$ and $d_i \in \mathcal{D}_{S_k}\}$). Similar strategy applies to $\mathcal{D}_D$.

Note that for $\chi^2 = 7.879$, there is only a probability of 0.005 that a wrong decision would be made such that a feature from a stationary process would be identified as not stationary, i.e. a random feature is wrongly identified as a significant feature. Hence, α is set to 7.879. Besides, only the features that appear in more than one-tenth of the documents in the corresponding period would calculate their χ^2 value. This is because rare features are difficult to estimate correctly and this can reduce significant computational cost. Algorithm 4 outlines the procedure of selecting positive training news stories.

Algorithm 4 select($\mathcal{D}$, $\mathcal{T}'_m$) – select positive training examples from a collection of documents $\mathcal{D}$ given a segmented time series $\mathcal{T}'_m$.

```
for each S_k in T'_m do
    L_{feature,k} = φ;
    D_{S_k} = φ;
    if t_rel(d_i) ≥ t_begin(S_k) and t_rel(d_i) < t_end(S_k) then
        Assign d_i to D_{S_k};
    end if
end for
for each S_k in T'_m do
    for each f_i ∈ D_{S_k} do
        if χ²(f_i) ≥ α then
            append f_i to L_{feature,k};
        end if
    end for
end for
D_R = φ;
D_D = φ;
for each f_j ∈ L_{feature,k} do
    if f_j ∈ {d_i | d_i ∈ D_{S_k}} then
        if slope(S_k) > 0 then
            Assign d_i to D_R;
        else
            Assign d_i to D_D;
        end if
    end if
end for
```

D. System Learning

Recall that in our training data, two types of data are available: $\mathcal{D}_R$ and $\mathcal{D}_D$. $\mathcal{D}_R$ and $\mathcal{D}_D$ represents the training documents correspond to the rise trend and the drop trend,6

respectively. Let N_R and N_D be the number of documents in $\mathcal{D}_R$ and $\mathcal{D}_D$, respectively. For the sake of simplicity, let us define $X \in \{R, D\}$.

Following the common practise of document preprocessing, for each document in X, $d_{j,X}$,a vector space model is constructed to represent it [33]:

$$d_{j,X} = \langle f_0 : w_{0,X}, f_1 : w_{1,X}, \ldots, f_n : w_{n,X} \rangle \quad (6)$$

where f_i is the i^{th} feature in $\mathcal{D}$ and $w_{i,X}$ is the weight of f_i in X. $w_{i,X}$ indicates the importance of f_i in $d_{j,X}$. Follow the existing works, we use a $tf \cdot idf$ schema for calculating the weights [36], [33]:

$$w_{i,X} = \begin{cases} tf_{i,j} \cdot \log_{N_X} \dfrac{N_X}{df_{i,X}} & \text{if } df_{i,X} \neq 0, \\ 0 & \text{if } df_{i,X} = 0. \end{cases} \quad (7)$$

where $tf_{i,j}$ is the *term frequency* (i.e. the number of times f_i appears in d_j) and $df_{i,X}$ is the *document frequency* (i.e. the number of documents contains f_i in X). Finally, $w_{i,X}$ is normalized to unit length so as to account for the differences in the length of each document.

In this formulation, each feature is regarded as a single dimension and the weight of the feature is regarded as the coordinate for that dimension. In other words, each document has n-dimension ($\Re^n$), where n is the total number of features in $\mathcal{D}$. Thus, our training data consists N_X pairs of $(d_{1,X}, y_{1,X}), (d_{2,X}, y_{2,X}), \ldots, (d_{N_X,X}, y_{N_X,X})$, with $d_{i,X} \in \Re^n$ and $y_{i,X} \in \{-1, 1\}$. This problem then reduced to a two class pattern recognition problem in which we are trying to find a hyperplane:

$$f : \mathcal{D}_X^T \beta_X + \beta_{0,X} = 0 \quad \text{and} \quad \| \beta_X \| = 1 \quad (8)$$

which maximize the *margin*, C_X, between the positive training examples in X and negative training examples in X. Thus, this problem reduced to the following optimization problem:

$$\max_{\beta_X, \beta_{0,X}, \|\beta_X\|=1} : \quad C_X \quad (9)$$

$$\text{subject to :} \quad y_{i,X}(d_{i,X}^T \beta_X + \beta_{0,X}) \geq C_X, \forall i \in X \quad (10)$$

By dropping the norm constraint on β_X, solving this problem is equivalent to solve the following optimization problem:

$$\min_{\beta_X, \beta_{0,X}} : \quad \| \beta_X \| \quad (11)$$

$$\text{subject to :} \quad y_{i,X}(d_{i,X}^T \beta_X + \beta_{0,X}) \geq 1, \forall i \in X \quad (12)$$

Since document vectors are very sparse, the two classes will share a high overlapping region in the feature space. To deal with it, slack variables, $\xi_X = (\xi_{1,X}, \xi_{2,X}, \ldots, \xi_{N_X,X})$, is introduced (Figure 5):

$$\min_{\beta_X, \beta_{0,X}} : \quad \frac{1}{2} \| \beta \|^2 + C \sum_{i=1}^{N_X} \xi_{i,X} \quad (13)$$

$$\text{subject to :} \quad y_{i,X}(d_{i,X}^T \beta_X + \beta_{0,X}) \geq 1 - \xi_{i,X}, \forall i \in X \quad (14)$$

$$\xi_{i,X} = 0, \forall i \in X \quad (15)$$

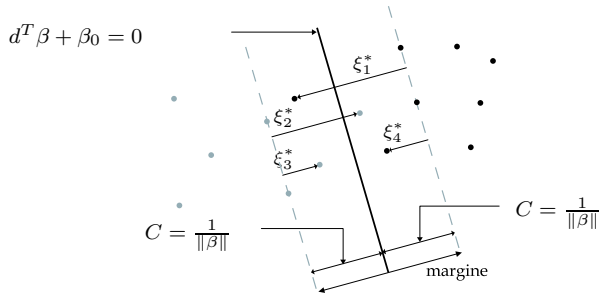


Fig. 5. The basic concept of the classifi er in a two-dimensional situation. The decision boundary is the solid line, while the broken lines bound the maximal margin of width $2C$. The points labeled ξ_j^* are on the wrong side of their margin by an amount $\xi_j^* = C\xi_j$. Points on the correct side have $\xi_j^* = 0$.

Constraint (14) requires that all training examples are classified correctly up to some slack $\xi_{i,X}$. If a training example lies on the wrong side of the hyperplane, the corresponding $\xi_{i,X}$ is ≥ 1. Therefore $\sum_{i=1}^{N_X} \xi_{i,X}$ is an upper bound on the number of training errors.

Consequently, solving this optimization problem is equivalent to solve a Support Vectors Machine (SVM) problem [37]. For computational reason, it is far more efficient to convert the above primal optimization problem to the Lagrangian (Wolfe) dual optimization problem [17], [37]:

$$\min : \quad L_X(\alpha) = -\sum_{i=1}^{N_X} \alpha_i + \frac{1}{2} \sum_{i=1}^{N_X} \sum_{j=1}^{N_X} Q_{ij} \quad (16)$$

$$\text{subject to :} \quad \sum_{i=1}^{N_X} \alpha_{i,X} y_{i,X} = 0 \quad (17)$$

$$Q_{ij} = \alpha_{i,X} \alpha_{j,X} y_{i,X} y_{j,X} x_{i,X}^T x_{j,X} \quad (18)$$

$$0 \leq \alpha_{i,X} \leq C, \forall i \in X \quad (19)$$

The size of the optimization problem depends on the number of training examples N. Defining a matrix $Q = y_i y_j x_i^T x_j$. Note that the size of Q is around N^2. For learning task with thousand of features and thousand of training documents, it becomes impossible to keep Q in memory. Standard implementations require either explicit storage of Q or re-compute Q every time when it is needed. However, this becomes prohibitively expensive. In this paper, we applied the technique describe by Joachims [38], which decompose the learning task into several sub-tasks. The solution of β_X, $\beta_{0,X}$ and $\xi_{i,X}$ can be computed as:

$$\beta_X = \sum_{i=1}^{N_X} \alpha_{i,X} y_{i,X} d_{i,X} \quad (20)$$

$$\beta_{0,X} = y_{sv,X} - \beta_X d_{sv,X} \quad (21)$$

$$\xi_X = \max\{1 - y_{i,X}(\beta_X d_{i,X} + \beta_{0,X}), 0\} \quad (22)$$

where the pair $(d_{sv,X}, y_{sv,X})$ must be a $support\ vector$ with $\alpha_{sv} < C$. Support vectors are those observations with the coefficient $\alpha_{i,X} \neq 0$.

TABLE II
THE CATEGORIES POWERED BY REUTERS.

Category	Category	Category
Hotel	Industry (A-G)	Consultant (A-G)
Financial	Industry (H-O)	Consultant (H-O)
Properties	Industry (P-Z)	Consultant (P-Z)
Utility	Germany	Miscellaneous

E. System Operation

After the system training process, two classification models are generated in which they are responsible for determining whether an unseen document would trigger the rise event and drop event, respectively. Given the solutions of β_X and $\beta_{0,X}$, the decision function for any unseen document, $\hat{d}$, can be written as:

$$\begin{aligned} G_X(\hat{d}) &= \text{sign}[f(\hat{d})] \\ &= \text{sign}[\hat{d}^T\beta_X + \beta_{0,X}] \end{aligned} \tag{23}$$

If $G_R > 0$ ($G_D < 0$), then the unseen document, $\hat{d}$ is classified as triggering the rise event (drop event). In other words, $\hat{d}$ is believed to affect the time series such that it goes upward (downward). If both G_R and G_D are < 0, then $\hat{d}$ is classified as noise which means that $\hat{d}$ does nothing with the time series. If both G_R and G_D are > 0, then the actual impacts of $\hat{d}$ is ambiguous since it triggers both the rise and drop events, which is impossible. In such a case, we would ignore $\hat{d}$ also, and classified it as noise as well.

IV. EVALUATION

A prototype system using Java™ is developed to evaluate the proposed system. All of the experiments are conducted on a Sun Blade-1000 workstation running Solaris 2.8 with 512MB physical memory and with a 750MHz Ultra-SPARC-III CPU. Intra-day stock prices and real-time news stories are archived through Reuters Market 3000 Extra[7] from 20^{th} January 2003 to 20^{th} June 2003. All data are stored into IBM DB2 Version 7.1[8].

For the real-time news stories, there are more than 350,000 documents archived. Note that Reuters has assigned to which sectors, countries, etc, the news stories should belong. Therefore, we do not need to worry about how these news stories should be organized. All features from the news stories are stemmed and converted to lower cases, in which punctuation and stop-words are removed, numbers, web page addresses and email addresses are ignored.

For the stock data, intra-day stock prices of all the Hong Kong stocks are recorded[9]. The stocks belong to one of the categories listed in Table II. According to the observations given by the technical analysis that price movements associated with light volumes denotes only temporal movements, but not trends, thus, for each stock, the transactions that are associated with light volumes (e.g. few hundred shares) are ignored. In order to account for the different price range of different stocks, stock prices of all stocks are normalized.

A. Time Series Evaluations

Figure IV-A shows the typical results after applying the t-test based split and merge segmentation algorithm on three stocks. Due to the space limited, we report only three cases. Other stocks behave in the similar way. The reported stocks are: 1) Cheung Kong (0001.HK); 2) Cathay Pacific (0293.HK) and 3) TVB (0511.HK). The unmodified stock data are shown on the top while the segmented data are shown on the bottom. We can see that the trends generated are quite reasonable and suitable. Note that the longest trend lasts for 2 weeks while the shortest one lasts for 3 days. This means that all of the trends generated are tertiary movements.

V. PREDICTION EVALUATIONS

One of the best way to evaluate the reliability of a prediction system is to conduct a market simulation which mimics the behaviors of investors using real-life data. As a result, two market simulations are conducted:[10]

- **Simulation 1: Proposed System**: Shares are bought or sold based solely on the content of the news stories. Two strategies are adopted:
  - For each stock, if the prediction of its upcoming trend is positive, then shares of it are bought immediately. The shares would be sold after holding for m day(s).
  - For each stock, if the prediction of its upcoming trend is negative, then shares of that stock are sold for short. The shares would be bought back after m day(s).

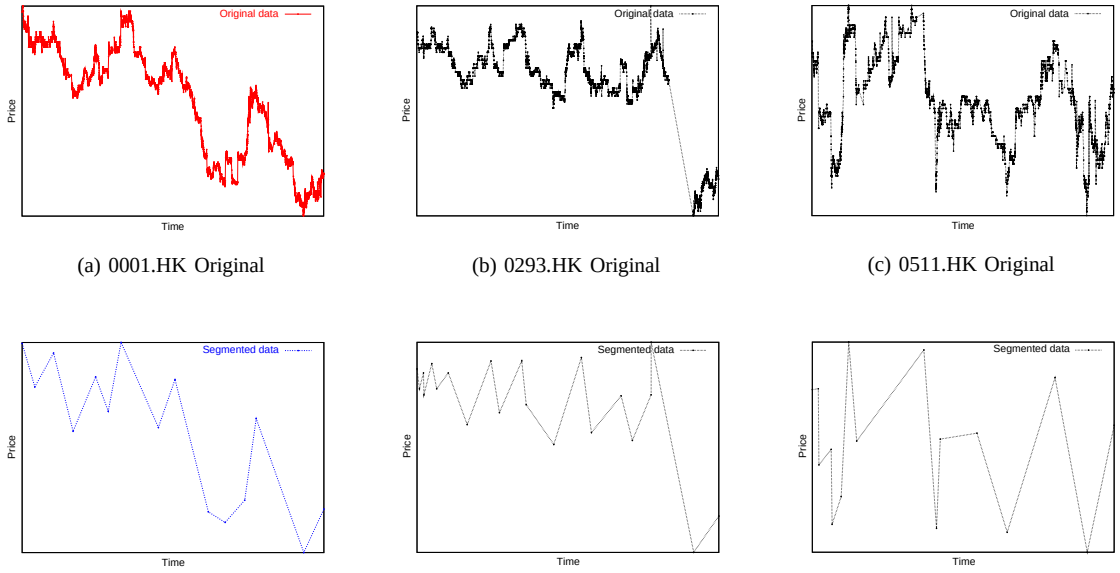
  An analysis of how m affects the evaluation results is given in Section V-B. In this section, m is set to 3 working days for simplicity. If the market is closed when the decision is made, then shares will be bought or sold in the beginning of the next active trading day.
- **Simulation 2: Buy-and-Hold Test**: For each stock, shares of that stock is bought at the beginning of the evaluation period. At the end of the evaluation period, all of the shares remain on hand are sold. This simulation serves as a base-line comparison which is used to demonstrate the *do-nothing strategy*.

In the above market simulations, rate of return, r, is calculated. As a result, how much shares are bought in each transaction could be ignored.

A. Simulation Results

Table III shows the results of the market simulations. From the table, Simulation 1 far outperforms Simulation 2. In order to see whether the earnings from the proposed system are statistically significant, another 1,000 simulations are conducted. In these simulations, the decisions of buying and selling were made at the same time as the proposed

[7] http://www.reuters.com
[8] http://www.ibm.com
[9] The stocks which have too few transaction records are ignored. This is simply because there are not enough data for training and/or evaluation.
[10] The assumption of zero transaction cost is carried out



(a) 0001.HK Original (b) 0293.HK Original (c) 0511.HK Original

(d) 0001.HK Segmented (e) 0293.HK Segmented (f) 0511.HK Segmented

Fig. 6. Before and after applying the t-test based split and merge segmentation algorithm. On the top: The original time series. On the bottom: The segmented time series. Three stocks are selected to report here: (1) Cheung Kong (0001.HK); (2) Cathay Pacifi c (0293.HK); and (3) TVB (0511.HK).

TABLE III

THE OVERALL EVALUATION RESULTS OF THE TWO MARKET SIMULATION. HERE, r IS THE RATE OF RETURN.

	Simulation 1	**Simulation 2**
Accumulative r	**18.06**	-20.56
Stand. Dev. of r	3.40	**2.15**
Maximum r	**12.42**	2.21
Minimum r	**-9.83**	-18.10
Top ten average r	**8.18**	1.11
Least ten average r	**-3.69**	-18.56

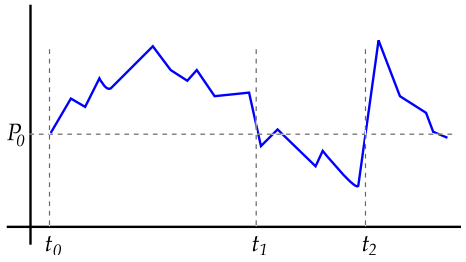


Fig. 7. A simple diagram illustrates the meaning of hit rate.

system, but without referencing to the contents of the news stories, i.e. the decisions are random. We then compare the cumulative earnings that are produced from the randomized trials and Simulation 1. For the randomized system, there are only 78 out of 1,000 trials that have rate of return exceed our proposed work. Thus, the proposed system is significant at the 0.5% level.

B. Hit Rate Analysis

Hit rate is another important measurement for the predictability of a forecasting system, especially for those kind of systems that are similar to our proposed one. Hit rate analysis can indicate how often the sign of return is correctly predicted. Figure 7 illustrates this idea. Assume that at t_0 a prediction which states that the stock price will go upward is made. Since from t_0 to t_1 (T_1), the stock prices are above p_0, we conclude that the prediction is correct in this period, i.e. *hit*. However, from t_1 to t_2 (T_2), the stock prices are below p_0, we therefore conclude that the prediction is wrong in T_2, i.e. *missed*. Thus, if the prediction period is varied, different conclusion could be drawn. In other words, the value of m in the market simulation presented in Section V-A is a critical factor.

TABLE IV

THE HIT RATE OF THE PROPOSED SYSTEM BY VARYING HOLDING PERIOD. HERE, THE RETURN IS CALCULATED IN RATE OF RETURN.

	Hit Rate	**Acc. Return**	**S.D. of Return**
1 day ($m = 1$)	51.0%	6.58	**1.147**
3 day ($m = 3$)	61.6%	18.06	3.400
5 day ($m = 5$)	**65.4%**	**21.49**	4.135
7 day ($m = 4$)	55.7%	7.22	3.791

Table IV shows the hit rate and the rate of return of the proposed system by varying the value of m. The accumulative return and hit rate both increase as m increase. It suggests that the system is most stable and suitable for applying the prediction within 3-5 days. It also suggests that such kinds of movements should be tertiary movements.

A careful examination of the prediction results would realize

that one of the major reasons for making error is that two news stories may be very similar in their contents, but have totally different implications.

VI. Conclusion

Scholars and professionals from different areas have shown that there is a high relationship between the news stories and the behaviors of the financial markets. In this paper, we revisit the problem and use real-time news stories and intra-day stock prices for our study. These data are chosen because they are readily available and the evaluation results obtained can easily be verified.

Several data mining and text mining techniques are incorporate in the system architecture. The tertiary movements on the stock price movements are identified by a novel piecewise linear approximation approach: a t-test based split and merge segmentation algorithm. News stories are aligned to the stock trends basic on the idea of Efficient Market Hypothesis. A document selection heuristics that is based on the χ^2 estimation is used for selecting the positive training documents. Finally, the relationship between the contents of the news stories and trends on the stock prices are learned through support vectors machine. Different experiments are conducted to evaluate various aspect of the proposed system. In particular, a market simulation using real-life data is conducted. Encouraging results are obtained in all of the experiments. Our study show that there is a high relationship between news stories and the movements of stock prices. Furthermore, by monitoring this relationship, actionable decisions could be made also.

References

[1] P. A. Adler and P. Adler, "The market as collective behavior," in *The Social Dynamics of Financial Markets*, P. A. Adler and P. Adler, Eds. Jai Press Inc., 1984, pp. 85–105.

[2] H. Blumer, "Outline of collective behavior," in *Readings in Collective Behavior*, 2nd ed., R. R. Evans, Ed. Chicago: Rand McNally College Pub. Co, 1975, pp. 22–45.

[3] J. M. Clark, "Economics and modern psychology," *Journal of Political Economy*, vol. 26, pp. 136–166, 1918.

[4] L. Tvede, *The Psychology of Finance*, revised ed. John Wiley and Sons, Inc., 2002.

[5] L. Festinger, *A theory of cognitive dissonance*. Stanford, Calif.: Stanford Univesity Press, Reprinted in 1968.

[6] M. Klausner, "Sociological theory and the bechavio of financial markets," in *The Social Dynamics of Financial Markets*, P. A. Adler and P. Adler, Eds. Jai Press Inc., 1984, pp. 57–81.

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

[8] ——, "Stock prediction: Integrating text mining approcah using real-time news," in *Proceedings of the 7th IEEE International Conference on Computational Intelligence for Financial Engineering*, Hong Kong, China, 2003, pp. 395–402.

[9] V. Lavrenko, M. D. Schmill, D. Lawire, P. Ogivie, D. Jensen, and J. Allan, "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, 2000, pp. 37–44.

[10] T. Fawcett and F. J. Provost, "Activity monitoring: Noticing interesting changes in behavior," in *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, San Diego, California, USA, 1999, pp. 53–62.

[11] D. Permunetilleke and R. K. Wong, "Currency exchange rate forecasting from news headlines," in *Proceedings of the 13h Australian Database Conference*, Melbourne, Australia, 2002, pp. 131–139.

[12] J. D. Thomas and K. Sycara, "Integrating genetic algorithms and text learning for financial prediction," in *Proceedings of the Genetic and Evolutionary Computing 2000 Conference Workshop on Data Mining with Evolutionary Algorithms*, Las Vegas, Nevada, USA, 2000, pp. 72–75.

[13] B. Wuthrich, D. Permunetilleke, S. Leung, V. Cho, J. Zhang, and W. Lam, "Daily prediction of major stock indices from textual www data," in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, New Youk, USA, 1998, pp. 364–368.

[14] R. J. Bauerm Jr and J. R. Dahlquist, *Technical Market Indicators*. John Wiley and Sons, Inc., 1999.

[15] R. D. Edwards and J. Magee Jr, *Technical Analysis of Stock Trends*, 5th ed. Springfield, 1966.

[16] S. A. Nelson, *The ABC of Stock Market Speculation*, 3rd ed. Fraser Publishing, 1903.

[17] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2002.

[18] F. Klein and J. A. Prestbo, *News and the Market*. Chicago: Henry Regenry, 1974.

[19] K. Nigam, J. Lafferty, and A. McCallum, "Using maximunm entropy for text classification," in *Proceeding of the 16th International Joint Conference Workshop on Machine Learning for Information Filtering*, Stockholm, Sweden, 1999, pp. 61–67.

[20] B. Wuthrich, "Probabilistic knowledge bases," *IEEE Transactions of Knowledge and Data Engineering*, vol. 7(5), pp. 691–698, 1995.

[21] ——, "Probabilistic knowledge bases," *International Journal of Intelligent Systems in Accounting Finance and Management*, vol. 6, pp. 269–277, 1997.

[22] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21th International Conference on Research and Development in Information Retrieval*, Melbourne, Australia, 1998.

[23] P. A. Adler and P. Adler, *The Social Dynamics of Financial Markets*. Jai Press Inc, 1984.

[24] W. J. Eiteman, C. A. Dice, and D. K. Eiteman, *The Stock Market*, forth ed. McDGraw-Hill Book Company, 1966.

[25] Y. Qu, C. Wang, and X. S. Wang, "Supporting fast search in time series for movement patterns in multiples scales," in *Proceedings of the 7th International Conference on Information and Knowledge Management*, Bethesda, Maryland, USA, 1998, pp. 251–258.

[26] C. Wang and X. S. Wang, "Supporting content-based searches on time series via approximation," in *Proceedings of the 12th International Conference on Scientific and Statistical Database Management*, Berlin, Germany, 2000, pp. 69–81.

[27] E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani, "An online algorithm for segmenting time series," in *Proceedings of the 1st IEEE International Conference on Data Mining*, San Jose, California, USA, 2001, pp. 289–296.

[28] T. Pavlidis and S. L. Horowitz, "Segmentation of plane curves," *IEEE Transactions on Computers*, vol. c23(8), pp. 860–870, 1974.

[29] R. O. Duda and P. E. Harts, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[30] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29(2-3), pp. 103–130, 1997.

[31] D. D. Lewis, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th International Conference on Research and Development in Information Retrieval*, Dublin, Ireland, 1994, pp. 3–12.

[32] ——, "The independence assumnption in information retrieval," in *Proceedings of the 10th European Conference on Machine Learning*, Chemnitz, Germany, 1998, pp. 4–15.

[33] F. Seabastiani, "Machine learning in automated text categorization," *ACM Computing Surverys*, vol. 34(1), pp. 1–47, 2002.

[34] W. B. Croft, "Boolean queries and term dependencies in probabilistic retrieval models," *Journal of the American Society for Information Science*, vol. 37(2), pp. 71–77, 1983.

[35] C. J. van Rijsbergen, "A theoretical basis for the use of co-occurrence data in information retrieval," *Journal of Documentation*, vol. 33(2), pp. 106–119, 1977.

[36] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Process Management*, vol. 24(5), pp. 513–523, 1998.

[37] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.

[38] T. Joachims, "Making large-scale svm learning practical," Computer Science Department, University of Dortmund, Tech. Rep. LS-8 (24), 1998.