# On the Construction and Analysis of Financial Time-Series-Oriented Lexicons

Chen-Yi Lai

*University of Taipei, Taipei, Taiwan*

*g10216005@go.utaipei.edu.tw*

Chuan-Ju Wang

*University of Taipei, Taipei, Taiwan*

*cjwang@utaipei.edu.tw*

Ming-Feng Tsai

*National Chengchi University, Taipei, Taiwan*

*mftsai@nccu.edu.tw*

## Abstract

This paper proposes a novel framework to build a time-series-oriented lexicon which can cover different types of sources and also has explicit links with the targets of prediction problems. In the framework, the input is composed of a text stream, such as financial news and a financial time series, such as the stock prices of a company. We then calculate the Pearson correlation between the frequency series of each word and the stock price series of a company. Although Pearson correlation gives a good idea of how much the two time series are correlated, it has a limitation in capturing the similarity when one of the series is stretched or shifted. To overcome this limitation, we adopt Dynamic time warping (DTW) to handle the problem. Finally, the words with high correlations will be extracted to build the time-series-oriented lexicon.

## 1. Introduction

Text analytics refers to the process of deriving high-quality information from textual information, and the corresponding techniques have been widely applied to many fields such as Biomedicine [1, 2] and Finance [3, 4]. In finance, there have been several studies that utilize textual analysis to examine the sentiments of news articles, financial reports, and user tweets about publicly-traded companies, and then use the examined sentiments to reflect the correlations with financial measures, such as stock returns and their volatilities [5, 6, 7, 8]. For most sentiment analysis algorithms, the sentiment lexicon is a very important resource and can be various in different fields [9]. In the field of finance, the first financial sentiment lexicon, which is proposed by Loughran and McDonald [7], has been widely used in several financial problems, such as financial risk prediction [10]. However, we believe that the financial sentiment lexicon has the following two limitations: First, the lexicon is constructed via only the 10-K financial reports, the wording of which is quite formal, thereby causing the problem that the words used in different sources, such as news articles and social networks, cannot be recognized. Second, the lexicon has no explicit link with the targets of prediction problems, such as stock prices and volatilities, which may cause the difficulty in analyzing the obtained prediction models.

To address these limitations, in this paper, we propose a novel framework to build a time-series-oriented lexicon that can cover different types of sources and also has explicit links with the targets of prediction problems, which should help us build a lexicon able to capture more target-oriented information. In the framework, the input consists of a text stream (e.g., the news from a time period) and a financial time series (e.g., APPLE stock prices or S&P 500); the output is a ranked list of words, in which the words are ranked by correlations of the word frequency series and the financial time series. Since the time series may be stretched or shifted, we adopt Dynamic Time Warping [11] to preprocess the two input time series. Finally, after obtaining the ranked list of words, we can construct a time-series-oriented lexicon according to the correlations. For instance, we construct a lexicon by including the words with the correlation less then $-0.5$ as the negatively correlated words and the ones with the correlation larger then 0.5 as the positively correlated words. In the preliminary experiments, we consider the S&P 500 index as the financial time series and the New York Times corpus as the text stream, which covers all the news published in the period of January 1, 1987 and July 19, 2007 including the total of over 1.8 million articles. As a direction of future work, we attempt to conduct a prediction task with the generated lexicon and also provide some analyses on both the lexicon and the resulting prediction models.

## 2. Methodology

This section first introduces the details of the proposed framework. The Pearson correlation and the Dynamic Time Warping (DTW) are also described.

### *2.1. Framework*

The main contribution of this paper is to propose a general framework to build a time-series-oriented lexicon. In the framework (see Figure 1), the input consists of a text stream (e.g., the news from a time period) and a financial time series (e.g., APPLE stock prices or S&P 500); the output is a ranked list of words, in which the words are ranked by the correlations of the word frequency series and the financial time series. Formally, given a news time series dataset $D = \{d_{t_1}, d_{t_2}, \ldots, d_{t_n}\}$ for the time point $t_1, t_2, \cdots, t_n$, we first calculate the word frequency series $WF_w = \{wf_{w,t_1}, wf_{w,t_2}, \ldots, wf_{w,t_n}\}$ for each word $w$, where $wf_{w,t_i}$ is the frequency of the word $w$ at the time $t_i$; the financial time series can be denoted as $TS = \{s_{t_1}, s_{t_2}, \ldots, s_{t_n}\}$. After calculating the correlation of $WF_w$ for each word $w$ and $TS$, we are able to obtain a ranked word list $WC = \{(w_1, c_1), (w_2, c_2), \ldots, (w_k, c_k)\}$, where $c_k$ denotes the correlation between the frequency series of word $k$ and the given financial time series. Finally, from the set of word list $WC$, we can construct a time-series-oriented

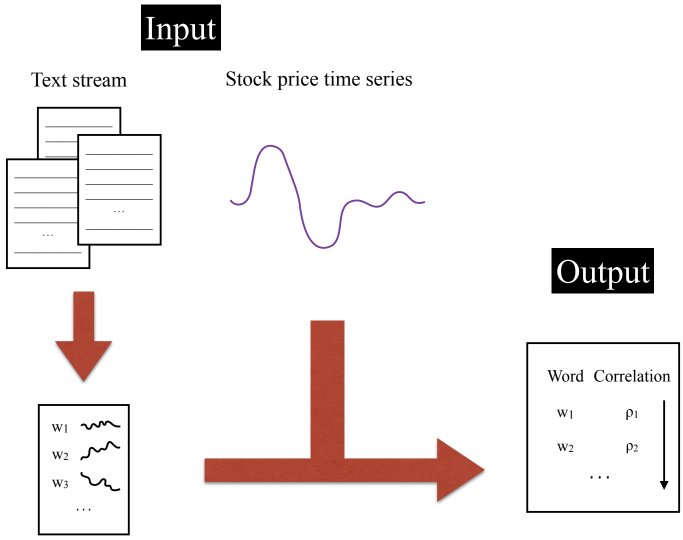

Figure 1: The Proposed Framework

lexicon according to the correlations $c_1, c_2, \ldots c_k$. For instance, we can construct a lexicon by including the words with $c_k < -0.5$ as the negatively correlated words and the ones with $c_k > 0.5$ as the positively correlated words.

### *2.2. Pearson Product-Moment Correlation Coefficient*

In statistics, the Pearson product-moment correlation coefficient is a measure of the linear correlation (dependence) between two time series $X$ and $Y$, which $X = \{x_1, x_2, \ldots, x_N\}$ and $Y = \{y_1, y_2, \ldots, y_N\}$, giving a value between $+1$ and 1 inclusive. In this paper, the variable $X$ can be the time series of stock prices of a certain company and $Y$ can be a word frequency series of a certain word. The Pearson correlation coefficient can be defined as follows:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y},$$

where $\mu_X$ and $\mu_Y$ are means of $X$ and $Y$, $\sigma_X$ and $\sigma_Y$ are standard deviations of $X$ and $Y$, $E$ is the expected value (average) operator, and $cov$ is the covariance operator. For example, Figure 2 plots the stock prices of `Apple` from January 1, 2004 to December 31, 2004 and Figure 3 plots the frequency series of the word "saint" in the same period; the resulting correlation between these two time series is around 0.146451.

### *2.3. Dynamic Time Warping (DTW)*

According to the low correlation obtained in the previous section, we believe that although Pearson correlation gives a good idea of how much the two time series are correlated,

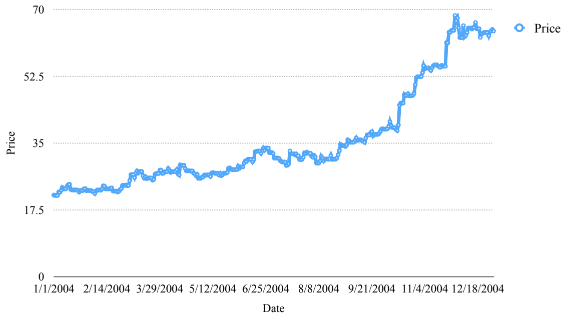

Figure 2: Stock Prices of `Apple` in 2004

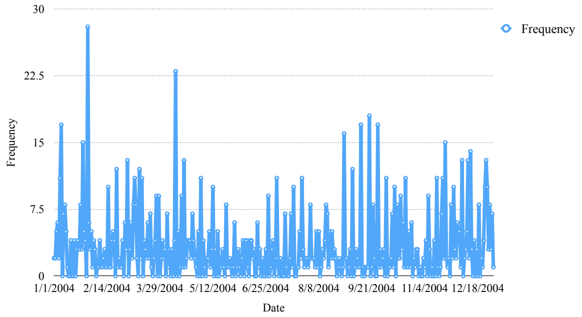

Figure 3: Word Frequencies of "Saint" in 2004

it still has a limitation in capturing the similarity when one of series is stretched or shifted (see Figure 4). It is often the case that two time series have overall similar shapes, but are not exactly lined up on the timeline. In reality, when one factor may affect another, there can be a delay in impact, or this impact may last longer even once the causal factor has disappeared. Therefore, to overcome this limitation, we adopt a technique called Dynamic Time Warping (DTW), which has been widely used in the filed of signal processing, to preprocess the time series.

DTW is a dynamic programming algorithm that aligns time seires with a flexible timeline mapping, which means depending on the shape of the series, a time period (a day in paper) is able to be dynamically mapped to several time periods days of the other series . By using the mapping path, DTW can find the best alignment path with the minimal distance between the two time series (e.g., Euclidean distance). DTW algorithm first builds the cost

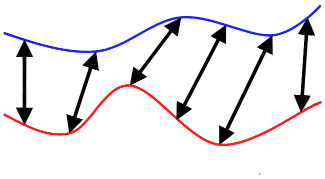

Figure 4: Dynamic Time Warping

matrix $C$ having distances for all pairs between the two time series $X$ and $Y$, in which each element is $c_{i,j} = ||x_i - y_i||$. With the cost matrix, we search an optimal alignment path, $p^*$, with a minimal:

$$\begin{aligned} DTW(X,Y) &= C_{p^*}(X,Y) \\ &= \min\{C_p(X,Y), p \in P^{N*M}\} \\ &= D(N,M), \end{aligned}$$

where $D$ is accumulated cost matrix, i.e., $D(i,j)$ is the minimal cost to align $x_1, ..., x_i$ to $y_1, ..., y_i$. $D$ can be computed with dynamic programming [11].

Figures 5 and 6 illustrate the resulting two time series of applying DTW to those in Figure 3 and 4; the correlation between these two processed time series becomes around 0.902166.

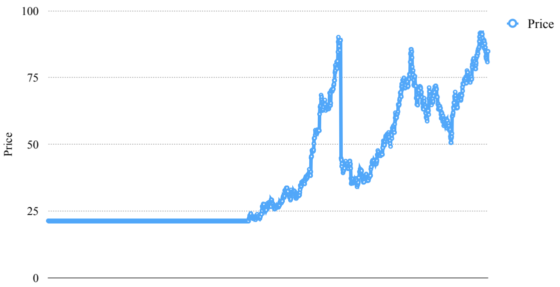

Figure 5: Stock Prices of `Apple` in 2004 After DTW

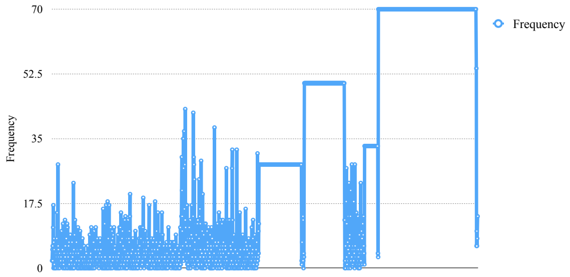

Figure 6: Word Frequencies of “Saint” in 2004 After DTW

## 3. Preliminary Experiments

### *3.1. Dataset*

In the experiments, we use the text stream obtained from the New York Times annotated corpus for the period from January 1, 2004 to December 31, 2006. Stock prices in time series are collected from the Wharton Research Data Services (WRDS) and the time period is set to the same as the text stream. We first investigate the four companies, `Apple`, `Amazon`, `Microsoft` and `Starbucks`, and their stock prices are used to build the four corresponding lexicons in the preliminary experiments.

### *3.2. Data Preprocessing*

#### *3.2.1. Text Indexing*

For the text preprocessing, we use Lemur[1] to build the index for the input text stream with the following two settings. First, we filter the stop words (such as a, an, and the), which refer to the most common words in a language. Second, the stemming technique is applied to reduce inflected (or derived) words to their word stem;[2] for example, the words, bought or buying, will be converted to their word stem“buy.” After building the index, the frequency of each word stem at each time point is calculated; therefore, we can obtain the frequency of each word stem in time series in a straight forward manner. After the above preprocessing, there are $368,509$ unique terms in total. To calculate the word frequencies on a daily basis, we integrate the texts from the same day to a new document; thus, there are $1,096$ documents in the 3 years.

[1] http://www.lemurproject.org/

[2] In the experiments, we adopt the Porter stemmer.

### *3.2.2. Dealing with Missing Data*

In the experiments, we collect the stock prices from WRDS. But it happens frequently that some observations in the middle of the sample are missing and they are denoted as NAs in the data. There are several ways to manage this problem, such as replacing, keeping, and removing. Both the keeping and removing approaches are not appropriate for our task since keeping the missing data with the NA or deleting the NAs (causing different lengths of the time series) make the correlation calculation infeasible. For the integrity of the data, we implement the replacing approach, which replace the missing data to the stock price in the previous day.

### *3.3. Experimental Results*

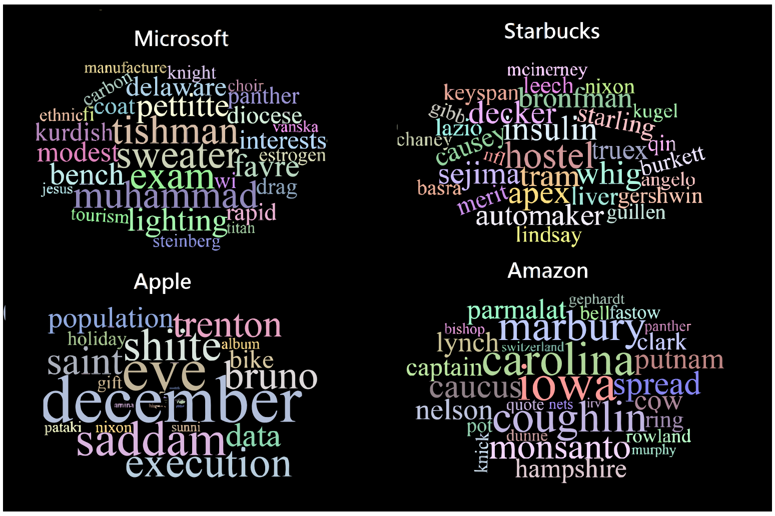

Figure 7: Word Cloud

Figure 7 plots the top ranked words of each company with a word cloud representation. From the figure, we observe that the words with higher correlations with the corresponding time series for the four companies are quite different. For example, the top three words for `Apple` are december, eve, and saddam, whereas those for `Microsoft` are sweater, tishman, and exam. The results show that the stock prices of different companies indeed generate different lexicons with the proposed approach.

## 4. Conclusion and Future work

This paper proposes a framework to construct a target-oriented lexicon, which contains all the words with highly correlations with the stock prices of a certain company. The preliminary results suggest that the generated lexicon for a certain company can vary from one another. For our future work, we will validate the results in the lexicons to predict the movements of stock prices, and, hopefully, the target-oriented lexicon can help us find more predictive models.

[1] S. Ananiadou, J. McNaught, Text mining for biology and biomedicine, Citeseer, 2006.

[2] T. Joseph, V. G. Saipradeep, G. S. V. Raghavan, R. Srinivasan, A. Rao, S. Kotte, N. Sivadasan, Tpx: Biomedical literature search made easy, Bioinformation 8 (12) (2012) 578.

[3] S. Kogan, D. Levin, B. R. Routledge, J. S. Sagi, N. A. Smith, Predicting risk from financial reports with regression, in: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2009, pp. 272–280.

[4] M.-F. Tsai, C.-J. Wang, Risk ranking from financial reports, in: Advances in Information Retrieval, Springer, 2013, pp. 804–807.

[5] A. Devitt, K. Ahmad, Sentiment polarity identification in financial news: A cohesion-based approach, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, 2007, pp. 984–991.

[6] D. Garcia, Sentiment during recessions, The Journal of Finance 68 (3) (2013) 1267–1300.

[7] T. Loughran, B. McDonald, When is a liability not a liability? textual analysis, dictionaries, and 10-ks, The Journal of Finance 66 (1) (2011) 35–65.

[8] S. M. Price, J. S. Doran, D. R. Peterson, B. A. Bliss, Earnings conference calls and stock returns: The incremental informativeness of textual tone, Journal of Banking & Finance 36 (4) (2012) 992–1011.

[9] R. Feldman, Techniques and applications for sentiment analysis, Communications of the ACM 56 (4) (2013) 82–89.

[10] C.-J. Wang, M.-F. Tsai, T. Liu, C.-T. Chang, Financial sentiment analysis for risk prediction, in: Proceedings of the Sixth International Joint Conference on Natural Language Processing, 2013, pp. 802–808.

[11] H. D. Kim, D. Nikitin, C. Zhai, M. Castellanos, M. Hsu, Information retrieval with time series query, in: Proceedings of the 2013 Conference on the Theory of Information Retrieval, 2013, pp. 14:56–14:63.