Financial Forecasting Using Time Series and News

Haizhou Qu

Doctor of Philosophy

University of York

Computer Science

February 2017

Dedication

To Love and Peace

Abstract

This thesis focuses on the field of financial forecasting. Most studies that use the financial news as an input in the prediction process, take it for granted that news has an effect on financial markets. The starting point for this research is the need to question this assumption, and if confirmed, to attempt to quantify it.

Therefore, the first study investigates the correlation between news and stock performance based on a dataset covering both trading data and news of 25 companies. We propose a novel framework to quantify the relationship based on two matrices of pairwise distances between companies. The first matrix represents distances between sets of news articles, while the other represents the pairwise distances between the financial performances. The detected correlation varies with time and reaches statistically significant. The next study focuses on testing if news can be used as a proxy for future financial performance in a profitable trading strategy. The one proposed here uses our previous findings to select the stock for which news affects most strongly on financial performance. The results show that this strategy outperforms competitive baselines.

Based on the proposed framework, a textual feature ranking method is proposed. This method assigns weights for textual features, and those weights are optimised to maximise the value of the relation to be quantified. A gradient descent algorithm is applied to obtain the optimal weights. There are two findings: first, named entity related words are weighted more than other words. second, optimal weights lead to a significantly better indicator for selecting winner stocks hence better profitable strategies.

Lastly, the popular convolutional neural work is used to implement a novel financial forecasting approach, which uses the stock chart as input. The results show that this approach can provide effective predictions of future stock price movements.

Contents

Abstract **3**

List of figures **7**

List of tables **13**

Acknowledgements **19**

Declaration **21**

Abbreviations **23**

Notations **25**

1 Introduction **27**

1.1 Motivation 27

1.2 Research Questions 28

1.3 Thesis Contributions 29

1.4 Thesis Organisation 29

2 Literature Review **31**

2.1 Financial Markets 31

2.2 Machine Learning Techniques 35

2.3 Natural Language Processing 42

2.4 Statistical Measures of Correlation 46

2.5 Related Work 49

3 Data Collection and Exploration **59**

3.1 Introduction 59

3.2 Data Sources 60

3.3 Parsing Online News 61

3.4 Explorations of Textual Data 63

4 Quantifying the Relationship between Financial News and Time Series **67**

4.1 Introduction 67

4.2 Overview of Framework 68

4.3 Distance Aggregation: News and Time Series 70

4.4 Relationship Quantifying: Mantel Test and Spearman's Rank-Order Correlation 75

4.5 Description of Data and Experiment 80

4.6 Results and Discussions 87

4.7 Conclusions 95

5 Learning Optimal Weights of Text Features for Financial Forecasting **97**

5.1 Introduction 97

5.2 Methodology: Feature Selection by Optimising Quantified Relation 98

5.3 Experimental Design 101

5.4 Results and Discussion 103

5.5 Conclusion 105

6 Financial Forecasting Based on Stock Charts **107**

6.1 Introduction 107
6.2 Related Work 108
6.3 Methodology 112
6.4 Results and Discussion 118
6.5 Conclusion 119

7 Conclusions **123**
7.1 Overview of Findings and Discussion 123
7.2 Limitations and Future Work 125

A Integrating Time Series with Social Media Data in an Ontology for the Modelling of Extreme Financial Events **127**

B Quantifying Correlation between Financial News and Stocks **135**

References **141**

List of Figures

2.1 Structure of a candlestick chart (opening higher than closing). 33

2.2 Candlestick chart of IBM stock from Jan. 1, 2015 to May 24, 2015. 34

2.3 Diagram of a Feed Forward Neural Network with one hidden layer. 36

2.4 Differences in hidden layers in FPNN and RNN 37

2.5 Unfolded recurrent hidden layer 37

2.6 Illustration of the concept of sparse connectivity. Each circle represents a neuron and each arrow represents a connection between neurons (Lab, 2017) . . . 38

2.7 Illustration of the concept of shared weights, each circle represents a neuron and each arrow represents a connection between neurons (Lab, 2017) 40

2.8 Example of a convolutional layer with four filters (channels) (Lab, 2017) 40

2.9 Plot of activate functions, $sigmoid(x)$ in blue, $tanh(x)$ in red and $relu(x)$ in green. 40

2.10 The structure of AZFinText. 50

2.11 The structure of NewsCATS. 51

2.12 The structure of the SVR + ARIMA forecasting model. 52

2.13 The structure of the event-driven forecasting model by Fung et al. (2005). . . . 54

2.14 The structure of a generalised textual financial forecasting system. 56

3.1 Structure of Yahoo Finance News RSS extraction system 61

3.2 Structure of pre-processing system 64

3.3 Logarithms of total number of news articles for stocks, ranked from high to low. 64

3.4 24 hour distributions of the number of financial news articles published in Yahoo Finance RSS feeds, for AAPL and GOOG, two NASDAQ listed stocks (data from 2013-1-28 to 2013-4-28). Market trading time for NASDAQ is between 9:30am & 4:00pm EST. 65

3.5 Weekly distributions of the number of financial news articles published in Yahoo Finance RSS feeds, for AAPL and GOOG, two NASDAQ listed stocks (data from 2013-1-28 to 2013-4-28). Market trading time for NASDAQ is between 9:30am & 4:00 pm EST. 66

4.1 Illustration of two implicitly defined spaces: $space^{\P}$ and $space^{\$}$. The solid lines refer to the distance between two stocks, GOOG (stock symbol of Google Inc) and MSFT (stock symbol of Microsoft Inc). 69

4.2 An overview of the proposed framework. Blue rectangles refer to input, output or intermediate data. Red rectangles stand for the key components of the framework. The framework produces a value ρ for given stock i and j for sliding window at time t. Depending on the Relationship Quantifier, the ρ may come with a *p-value* to indicate if ρ is statistically significant or not. 69

4.3 An OOV example sentence, where `taptic` is the OOV word. The contexts are `engine`, `det_a`, `case_for`, `nmod for_inv_asked` when querying vectors from the pre-trained Word2Vec model. 71

4.4 Illustration of the document mover's distance between two document sets. . . . 73

4.5 Illustration of *permutation* operation on matrix $\boldsymbol{X}$. The permutation randomly switches object i and j so that corresponding elements in the distance matrix switch positions accordingly. 76

4.6 A pair of sliding windows at time t: The time axis represents trading days over time. The sliding window of news in blue has $w^{¶}$ days and the sliding window of the time series includes $w^{\$}$ trading days. The two sliding windows do not overlap. Every pair of sliding windows goes one trading day ahead of the previous pair. 80

4.7 Mantel correlations ρ_M of time series. The red solid line and diamonds refer to tests on distance based on W2Vdep , the green solid line and squares refer to those based on W2V$words$ and the blue solid line and triangles refer to those based on TFIDF 87

4.8 Spearman correlations ρ_S of time series. The red solid line and diamonds refer to tests on distance based on W2Vdep , the green solid line and squares refer to those based on W2V$words$ and the blue solid line and triangles refer to those based on TFIDF 88

4.9 Comparsion of financial performances of trading strategies. 92

4.10 Accumulative Portfolio Values (APV) of strategies using the Mantel test as Relationship Quantifier. The top figure shows those strategies using Word2Vec for the Mantel test and the bottom figure shows those using TFIDF . Both figures contain the baseline strategies, BS (dashed line) and UCR (dotted line). 93

4.11 Accumulative Portfolio Values (APV) of strategies using Spearman's Rank-Order Correlation Coefficient asthe Relationship Quantifier. The top figure shows those strategies using Word2Vec and the bottom figure shows those using TFIDF . Both figures contain the baseline strategies, BS (dashed line) and UCR (dotted line). 94

5.1 Illustration of our approach. Grey dots represent vectors of stocks after optimisation. In which case, the pair-wised distances (AAPL-MSFT, AAPL-GOOG, etc.) will produce higher quantified relationship than those without optimisation (black dots). 99

5.2 Distribution of optimised weights of unique words, ranked from low to high. . 103

5.3 Accumulative Portfolio Values (APV) of strategies. The curves after the 103rd day (black vertical line) are in the test range in which the Word2Vec splits into three branches: Word2Vec , Word2Vec +TFIDF , Word2Vec +$\boldsymbol{W}_{optimal}$. . . . 104

5.4 Accumulative Portfolio Values (APV) of strategies (zoomed in). The curves after the 103rd day (black vertical line) are in the test range in which the Word2Vec splits into three branches: Word2Vec , Word2Vec +TFIDF , Word2Vec +$\boldsymbol{W}_{optimal}$. . 104

6.1 Illustration of dropout mechanism from the original dropout paper (Srivastava et al., 2014). The left figure shows a standard neural network with two hidden layers. The figure on the right shows the network with applied dropout technique during training. Crossed units are the dropped out nodes. 109

6.2 Figure from (Long et al., 2015). It illustrates fully connected layers. The cat picture above represents the receptive field in the input image. 111

6.3 Examples of candlestick charts as training data, generated from IBM minute level `OHLCV`. 113

6.4 Examples of scaled trading data as training data, generated from IBM minute level `OHLCV`. Darker pixels represent higher values. 114

6.5 Histogram plot of *return* of daily and minute `OHLCV` data of IBM stock. . . . 114

6.6 Structure of FCN1D network. Conv1D refers to a one dimensional (1D) convolutional layer. 116

6.7 Structure of FCN2D network. Conv2D refers to 2D convolutional layer. . . . 117

6.8 Illustration of ResNet50 structure (Malhotra, 2017) by stages. 117

6.9 To buy or not to buy? This figure shows an example of directional forecasting causing an extremely risky investment decision. Assume a perfect forecasting algorithm gives the probability of $p(c_{t+1})$ price change c in future at time $t+1$. c_t is price change at time t, $P[a \leq c_{t+1} \leq b]$ is the probability that the price will change between level a and b, e.g. $P[3\% \leq c_{t+1} \leq 4\%] = 15\%$. The most likely is that the price will drop 5% or even more. However if using the directional approach, the probability of going up is summed up to 55%, higher than the probability of dropping down of 33%. The detailed information of risk distribution is hidden. 121

List of Tables

2.1 Example of stock trading order book 33

3.1 Different textual data sources for finance. There are five metrics to measure: Co* = Comprehensiveness, Ex* = Expertise, Im* = Immediacy, Ac* = Accuracy, Av* = Availability. H = High, M = Medium, L = Low. 60

4.1 Overview of data used in this research 81

4.2 Information for each company: symbol, name, stock exchange, close price (ADJ) and number of news items per day. 81

4.3 The Pearson's Correlation results between next day's intraday return r and δ defined in section 4.5.3 across all stocks. Each number is the portion of trading days when δ and r have a significant linear correlation with a confidence level of 95%. 90

4.4 Ratios when selected stock are in next day's top five winners. Each number is the portion of trading days when the selected stock appears in the top five stocks with highest intraday return on the next trading day. 90

4.5 Financial performances of trading strategies. There are three groups of participants. The first group shows results using Mantel's Correlation as the Relationship Quantifier; the second group shows results using Spearman's Rank-Order Correlation as the Relationship Quantifier; in the last group BS and UCR are baselines, where BS is the Best Stock strategy, and UCR is the Uniform-Constant Rebalance strategy. fAPV refers to final-Accumulated Portfolio Value; SR stands for Sharpe Ratio; MD is Maximum Drawdown; AR is the Annualised Return. Bold numbers labeled with '*' are the best among the participants. . . . 91

5.1 Dataset split for experiment. Note that dates are inclusive on the left and exclusive on the right. 102

5.2 Sampled terms from top, middle and bottom in the ranking with regards to optimal weights. 104

5.3 Financial Performances of trading strategies during test date ranges. There are three groups of participants. The first group shows results using normal Word2Vec for text representation; the second group shows results using TFIDF and learned optimal weights; in the last group BS and UCR are baselines, where BS is the Best Stock strategy, and UCR is the Uniform-Buy-and-Hold strategy. fAPV refers to final-Accumulated Portfolio Value; SR stands for Sharpe Ratio; MD is Maximum Drawdown; AR is the Annualised Return. Bold numbers are the best participants. 105

6.1 Sample distribution of the training data. 115

6.2 Details of training, validation and test dataset used (all from IBM stock). 117

6.3 Hyper-parameters for Network Training 118

6.4 Result of performances of FCN1D, FCN2D, ResNet50 on **minute interval** stock data. All evaluated using test samples. DA refers to Directional Accuracy, P for Precision, R for Recall, F1 for F1-Score and Support is number of samples with the specified label. 119

6.5 Result of performances of FCN1D, FCN2D, ResNet50 on **minute interval** stock data. All evaluated using test samples. DA refers to Directional Accuracy, P for Precision, R for Recall, F1 for F1-Score and Support is number of samples with the specified label. 119

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Kazakov for his continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me throughout the time of this research and the writing of this thesis.

Declaration

I declare that the research described in this thesis is original work, which I undertook at the University of York during 2012 - 2017. Except where stated, all of the work contained within this thesis represents the original contribution of the author.

Some parts of this thesis have been published in conference proceedings and journals; where items were published jointly with collaborators, the author of this thesis is responsible for the material presented here. For each published item the primary author is the first listed author.

1. Qu, H. and Kazakov, D. (2016). Quantifying correlation between financial news and stocks. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6, Athen, Greece

2. Qu, H., Sardelich, M., Qomariyah, N. N., and Kazakov, D. (2016). Integrating time series with social media data in an ontology for the modelling of extreme financial events. In *LREC 2016 Joint Second Workshop on Language and Ontology & Terminology and Knowledge Structures*, Portoroz, Slovenia

    - Marcelo Sardelich: Twitter data parsing, polarity keywords extraction.
    - Nunung Nurul Qomariyah: Ontology design.

Abbreviations

ANN	Artificial Neural Network
SVM	Support Vector Machine
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
FFNN	Feed Forward Neural Network
ReLU	Rectifier Linear Unit
PoS	Part-of-Speech
BoW	Bag-of-Words
OHLCV	Open High Low Close Volume
TFIDF	Term Frequency - Inverse Document Frequency
RL	Reinforcement Learning
DL	Deep Learning
EST	Eastern Summer Time

Notations

x	A scalar
$\boldsymbol{x}$	A vector or time series
$\boldsymbol{X}$	A matrix
$\mathbf{X}$	A tensor
$\mathbb{X}$	A set
x_i	Element i of $\boldsymbol{x}$, index starts from 1
$X_{i,j}$	Element i, j of $\boldsymbol{X}$
$X_{i,:}$	Row i of $\boldsymbol{X}$
$X_{:,i}$	Column i of $\boldsymbol{X}$
$\bar{x}$	mean of $\boldsymbol{x}$
$\circ$	Hadamard's operator that applied to element-wise
$\boldsymbol{A} \odot \boldsymbol{B}$	Element-wise product of $\boldsymbol{A}$ and $\boldsymbol{B}$
$\tilde{x}$	a random variable, $\sim$ indicates the variable is random

Chapter 1

Introduction

This thesis focuses on the field of financial forecasting, which is to estimate or predict the future financial outcomes of financial assets. This field is challenging since the data generated by financial markets is mostly fuzzy, noisy and non-stationary. The development of Natural Language Processing techniques have enabled researchers to incorporate textual data, in addition to a financial time series, for the purpose of making better predictions.

This thesis addresses the need to demonstrate convincingly the relevance of news to financial forecasting and to propose ways to measure the magnitude and statistical significance of that relevance. In addition, it contributes to the field of financial forecasting through the development of an original dataset and a novel data representation for a neural network-based forecasting technique.

1.1 Motivation

Financial forecasting can be defined as an estimate of future financial outcomes for a financial asset using historical data. It is an important task since the financial markets are compulsory subsystems of modern economies. Predictions are a critical reference when a decision has to be made by investors, or even a government.

A number of studies apply natural texts such as Twitter data (Wolfram, 2010; Arias et al., 2014; Si et al., 2013; Bollen et al., 2011; Ruiz et al., 2012; Qu et al., 2016), financial news (Gidofalvi and Elkan, 2001; Boudoukh et al., 2013; Foucault et al., 2012; Ding et al., 2015), regularly published reports (Kogan et al., 2009) and Wikipedia usage (Moat et al., 2013), with varying degrees of success. After reviewing most of the existing studies, three noticeable gaps were discovered. Firstly, unlike typical natural language processing research, no gold standard dataset is available in this field. Therefore, researchers in this field have to compile their own datasets. Secondly, existing work is based on a strong 'active learning paradigm' that assumes that news can always be used for financial forecasting. However, there is no evidence that this assumption holds all the time. Thirdly, most existing studies model financial trading data in the same way that they process a time series. However, the conventional way of human traders relying on stock charts to make accurate predictions remains unexploited.

1.2 Research Questions

This thesis focuses on the following research questions:

1. Is news ever useful in financial forecasting on the time scale of one or more days?

2. If there is a relationship between news and the state of the market, how can we detect and quantify it?

3. What choice of representation of the news will make the correlation between news and stocks most evident?

4. Can one employ a novel, two-dimensional, representation of a financial time series to successfully train a convolutional neural network for the prediction of time series?

1.3 Thesis Contributions

This section lists the contributions of the thesis ranked by importance:

1. A novel framework that is able to identify and quantify the relationship between financial news and a financial time series. Preliminary work (see appendix B) is published in (Qu and Kazakov, 2016).

2. An effective indicator to help select a winning asset based on the first contribution.

3. A feature ranking method based on the first contribution by optimising the weights of words with regard to maximising the quantified relationship.

4. Exploration of using stock charts as input and approaching the financial forecasting problem using a vision-based deep learning network.

5. A dataset that integrates textual data and financial data using an ontological style to describe a catastrophic financial event (appendix A).

1.4 Thesis Organisation

This thesis is organised as follows:

Chapter 2: Literature Review This chapter starts with an introduction to financial markets. It then provides the relevant details of the Machine Learning and Natural Language Processing techniques used in this thesis, as well as knowledge related to the Mantel test. Finally, it gives an overview of related research on financial forecasting using textual data.

Chapter 3: Data Collection and Exploration This chapter presents the detail concerning the collection of the dataset used in this research. It covers selecting the data sources, implementation of data collection, as well as explorations of the collected textual data. It also

presents contribution 4 which integrates textual data and financial data in an ontological way to describe a catastrophic financial event.

Chapter 4: Quantifying the Relationship between Financial News and Time Series This chapter presents contribution 1. It introduces an innovative framework that uses the Mantel test to identify and quantify the correlation between news and a financial time series with respect to a set of stocks. The results are obtained using two approaches. It shows that research question 2 has been answered. An approach called *leave-one-out* using causal links is exploited to identify which stock has the greatest potential to be traded. An evaluation based on the Wilcoxon test to verify the stocks selected by *leave-one-out* also shows the effectiveness of causal links.

Chapter 5: Learning Optimal Weights of Text Features for Financial Forecasting This chapter describes a novel feature selection method based on the framework introduced in Chapter 4. It begins with a brief introduction and motivation. This is followed by a detailed algorithm and implementation. The results show interesting findings and provide supportive evidence for the effectiveness of the method.

Chapter 6: Financial Forecasting Based on Stock Charts This chapter proposes a novel method based on a convolutional neural network which is able to accept stock charts as input and make predictions like human traders do. The chapter begins with an introduction that explains the motivation. A description of how the data is prepared and the structure of the neural network is then presented. A number of results are shown to present the potential use of this method.

Chapter 7: Conclusions The beginning of this chapter summarises the thesis. The contributions of the thesis are then revisited. The limitations of the study and future work are discussed in the final part of the chapter.

Chapter 2

Literature Review

2.1 Financial Markets

2.1.1 Efficient Market Hypothesis

The Efficient Market Hypothesis (EMH) originated with Hayek (1945) and was further developed and improved by Fama (1970). It is considered to be one of the seven most important theories in Finance. The definition of an efficient market is that asset prices reflect all available information. The EMH can be described formally using Equation 2.1.

$$E(\tilde{P}_{t+1}|\phi_t) = (1 + E(\tilde{r}_{t+1}|\phi_t)) \cdot p_t \tag{2.1}$$

where E is the expectation, p is the price of an asset in the market, t is timestamp, r is the percentage return, ϕ is information set and $\sim$ indicates that the variable is random.

From Equation 2.1 we can see that the expected asset price at time $t + 1$ is based on the price at time t being proportional to expected return $\tilde{r}_{t+1}$ given ϕ_t. This implies that given the available information at t, the price in the future is not predictable.

In practice, to determine whether a security market is efficient depends on two aspects:

whether or not security prices adjust accordingly with relevant information updates, and whether relevant information can be delivered to all market participants equally, with the same immediacy and quality. If a security market satisfies the two assumptions above, according to the EMH, it is an efficient market. Security prices in this market are not predictable, since the prices already reflect all available information and no further information can be obtained using methods like technical analysis.

These two premises were not established at the time. There have been a number of studies in this area (Butler and Kazakov, 2012; Basu, 1977; Chan et al., 1997) providing exceptions when the EMH does not stand up well.

2.1.2 Stock Price Data

On the stock market, stock trading data are generated on the stock exchange. For a given stock, the market status can be represented as a table called the 'order book'. Quotations are divided into the sell side and the buy side, and for each side there are two columns, shares and bid/ask price correspondingly. Table 2.1 shows an order book that has 5-depth information. Here the depth means the price step for each side. The lowest ask says that there are quotes looking to sell 1000 shares of stock at the price $10.1. Note that those 1000 share quotes can come from different market makers. Once there is a match between bidding and asking prices or a market order is executed[1], the stock price will change to the deal price. For example, in 2.1, if there is a quote looking to buy 200 shares at $10.1, the stock price will be changed to $10.1, and the 1000 shares on the sell side (Ask) will become 800 shares, with 600 on the buy side (Bid).

Depending on how much information is revealed, two types of real-time trading data are usually provided by stock exchanges:

Level I Real-time highest bid with lowest ask quotes or last executed order price.

[1] A type of order that is immediately executed at the best available price.

Table 2.1: Example of stock trading order book

Shares	Bid	Ask	Shares
		\$10.6	5000
		\$10.5	2000
		\$10.4	3000
		\$10.2	600
		\$10.1	1000
800	\$10.0		
500	\$ 9.9		
3000	\$ 9.8		
2000	\$ 9.7		
5500	\$ 9.6		

Level II Real-time bid and ask quotes over different depths, also the market makers are revealed in most stock exchanges.

Traders use stock charts to understand stock data, since they are intuitive and it is easy to see patterns in them. The candlestick chart is the most widely used chart for visualising stock price. In a candlestick chart, each bar represents a time frame, for example, if a day is the time frame, we get a daily candlestick chart. The candlestick consists of a box and a pin, where the bottom and top of the box are determined by the stock price at the beginning (open) of the time frame and at the end (closing). The bottom and top of the pin are determined by the highest (high) and lowest (low) prices during the time frame. In order to distinguish whether the closing price is higher than the opening, the candlesticks will be displayed in two different colours, usually green and red.

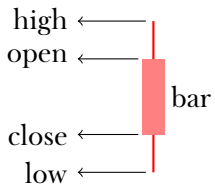


Figure 2.1: Structure of a candlestick chart (opening higher than closing).

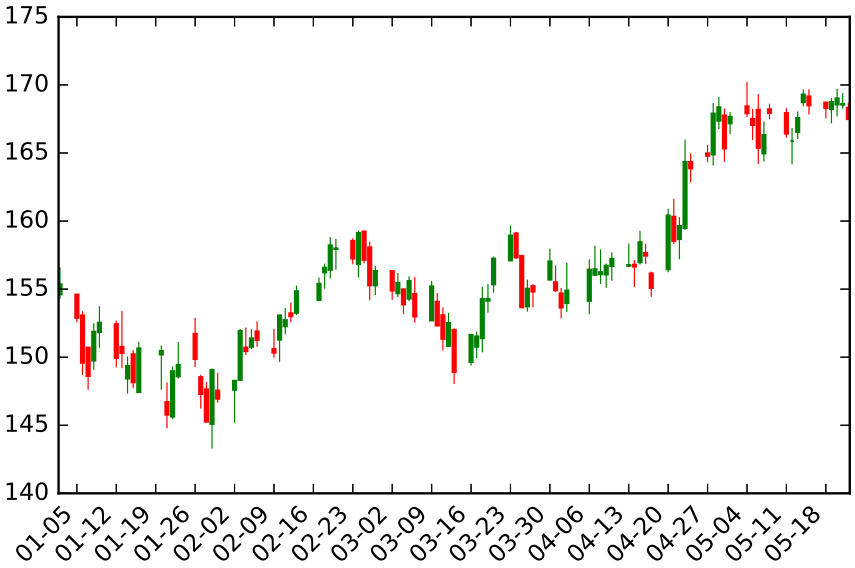


Figure 2.2: Candlestick chart of IBM stock from Jan. 1, 2015 to May 24, 2015.

2.1.3 Fundamental Analysis

Fundamental analysis is a type of security evaluation method that analyses financial reports, relevant news and the macroeconomic environment to determine or forecast the price of one or more securities. For example, to determine the price of a company stock, the following factors are commonly involved in the fundamental analysis: global and domestic economic analysis (GDP, inflation rate, forex rates, energy prices, etc.); industrial analysis (upstream-downstream industries); and, company analysis (product sales, market performance, customer analysis).

It is difficult to cover all information and collected information could be biased, inaccurate or even fake. The analysis can be subjective and highly related to a person's professional experience, which may cause unreliable outcomes. As only human experts are able to judge, collect and analyse, fundamental analysis usually takes a long time. Since the market is highly dynamic, there is a big risk that the result will not apply to the most recent market.

2.1.4 Technical Analysis

Technical analysis is a method that forecasts future stock prices using historical market data such as close price, trading volumes and even market sentiment. As introduced in Section 2.1.1, such techniques cannot extract any new information in a market that is believed to be efficient. In this case, traditional technical analysis appears to be one of the greatest gaps between academic finance approaches and industry practices. Traditional technical analysis is known as 'charting', that is, exploring stock charts and finding geometric shapes or patterns which may contain indicative information for future price trends. Since it lacks scientific evidence and convincing examples, some circles often have a prejudice against traditional technical analysis.

2.2 Machine Learning Techniques

2.2.1 Neural Networks

Artificial Neural Networks (ANN) are inspired by biological neuronal systems which are similar to the brain. The first artificial neuronal system was proposed in 1943 (McCulloch and Pitts, 1943). This information processing paradigm is usually composed of a number of neurons which are connected to each other by connections that are assigned weights to determine their importance. Those weights are considered to be the most important elements of a neural network. The weights can be learned from a training process that optimises the network by repeatedly adjusting the values of weights so that the network output is close enough to the target output.

2.2.1.1 Feed Forward Network

The simplest Neural Network is a Forward Propagation Neural Network (FPNN) consisting of three layers: the input layer, the hidden layer and the output layer. Figure 2.3 shows the

structure of an FFNN with one hidden layer.

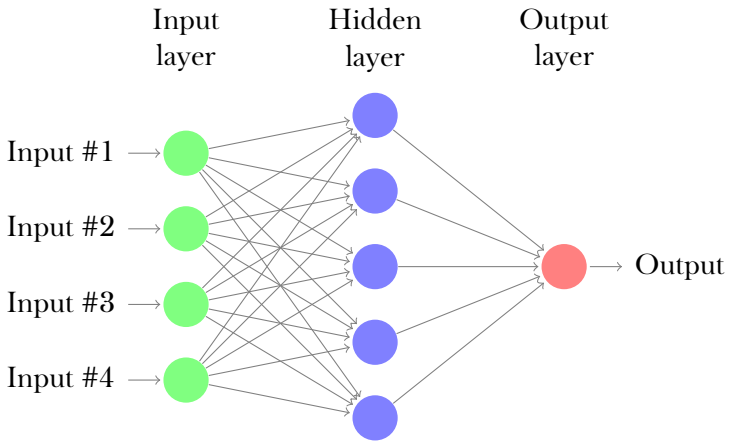


Figure 2.3: Diagram of a Feed Forward Neural Network with one hidden layer.

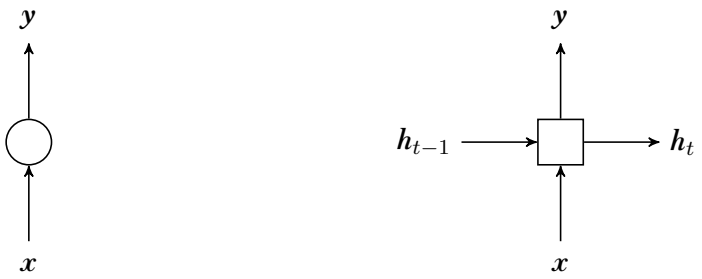
The hidden layer of FFNN is defined by Equation 2.2.

$$\mathbf{y} = f(\boldsymbol{W}\boldsymbol{x}) \tag{2.2}$$

where f is the activate function, $\boldsymbol{x}$ is the input vector and $\boldsymbol{y}$ is the output vector. It is illustrated in Figure 2.4a, where the circle represents an activate function.

2.2.1.2 Recurrent Networks

A recurrent network (RNN) aims to preserve the information of each forward step. Equation 2.3 describes one forward step of an Elman recurrent layer. It can be represented as in Figure 2.4b. Given a sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t$, the RNN unit can be unfolded with respect to time steps, as shown in Figure 2.5.



(a) Hidden layer of FPNN (b) Hidden layer of Elman recurrent network

Figure 2.4: Differences in hidden layers in FPNN and RNN

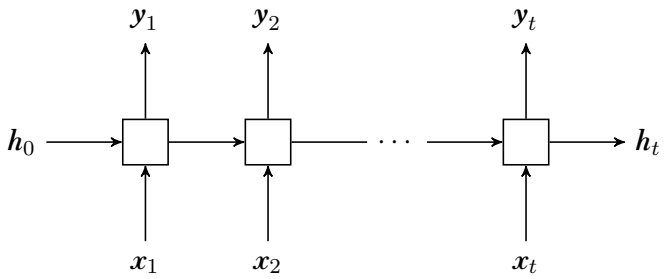


Figure 2.5: Unfolded recurrent hidden layer

$$\boldsymbol{h}_t = g(\boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{W}\boldsymbol{h}_{t-1}) \tag{2.3}$$

$$\boldsymbol{y}_t = f(\boldsymbol{V}\boldsymbol{h}_t) \tag{2.4}$$

where

f, g : activate functions.

$\boldsymbol{x}_t$: input at time t.

$\boldsymbol{y}_t$: output at time t.

$\boldsymbol{h}_t$: hidden state at time t.

$\boldsymbol{h}_{t-1}$: hidden state from previous time $t-1$.

$\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}$: weight matrices.

2.2.1.3 Convolutional Neural Network

Convolutional neural networks (CNN) are inspired by the animal visual cortex. In 1962, Hubel and Wiesel presented research on the visual cortex of the cat. The cells in the visual cortex are sensitive to smaller regions of the visual field (aka the receptive field). These cells are similar to local filters (channels) that are able to exploit the strong spatially local correlation in images. This is the most important feature of CNN, called local awareness. This mechanism of the visual cortex is approximated in the neural network by sparse connectivity (see Figure 2.6) and shared weights. The calculation of a convolutional layer is explained in Figure 2.8. Equation 2.5 gives the calculation for convolution operations.

$$h_{ij}^k = \delta((W^k * x)_{ij} + b_k) \tag{2.5}$$

where h^k is the k-th feature map, δ is the activate function (usually $ReLU$), W^k is weights of k-th feature map, and b^k is the bias for k-th feature map. $*$ is the convolution operator defined in Equation 2.6.

$$o[m,n] = f[m,n] * g[m,n] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f[u,v] \cdot g[m-u, n-v] \tag{2.6}$$

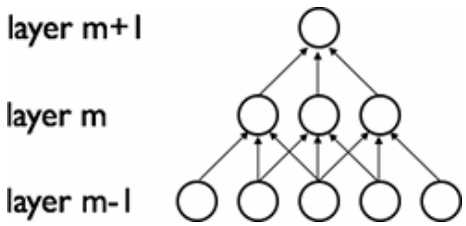


Figure 2.6: Illustration of the concept of sparse connectivity. Each circle represents a neuron and each arrow represents a connection between neurons (Lab, 2017)

.

2.2.1.4 Activate Functions

$Sigmoid$ (Equation 2.7) and $Softmax$ (Equation 2.8) are the two commonly used output activate functions. The former is used for binary classification and the latter for multi-classification. The output range, from 0 to 1, is the conditional probability $p(y = class|x)$. See Figure 2.9 for a plot of the $Sigmoid$ function

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

$$Softmax(\mathbf{x}) =< \frac{e^{x^{(k)}}}{\sum_{d=1}^{K} e^{x^{(d)}}} | k \in \{1, 2, \cdots, K\} > \tag{2.8}$$

where $\mathbf{x}$ is a K dimensional vector.

2.2.1.5 Loss Function

Mean Square Error (MSE) is a widely used loss function. However, for the $Sigmoid$ layer when $x > 6$ or $x < -6$, the gradients will be close to 0, which causes slow optimisation. For the $Sigmoid$ output layer, *Binary Cross Entropy* (BCE) or *Log Loss* function can be applied as a loss function, and for the $Softmax$ output layer, *Categorical Cross Entropy* (CCE) is a good choice as a loss function.

$$BCE(\mathbf{y}, \mathbf{y}') = -\sum_{d=1}^{D} (1 - \mathbf{y}) \odot log(1 - \mathbf{y}') + \mathbf{y} \odot log(\mathbf{y}') \tag{2.9}$$

$$CCE(\mathbf{y}, \mathbf{y}') = -\sum_{d=1}^{D} \mathbf{y} \odot log(\mathbf{y}') \tag{2.10}$$

where $\mathbf{y}$ is the target output and $\mathbf{y}'$ is the network output, both are D dimensional vectors, and operator $\odot$ means element-wise multiplication.

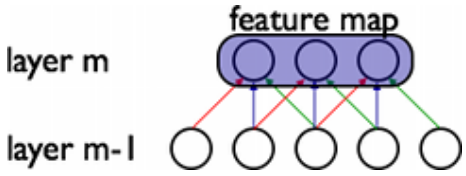


Figure 2.7: Illustration of the concept of shared weights, each circle represents a neuron and each arrow represents a connection between neurons (Lab, 2017)

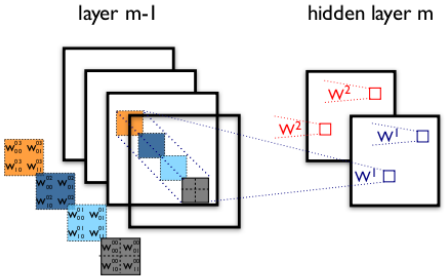


Figure 2.8: Example of a convolutional layer with four filters (channels) (Lab, 2017)

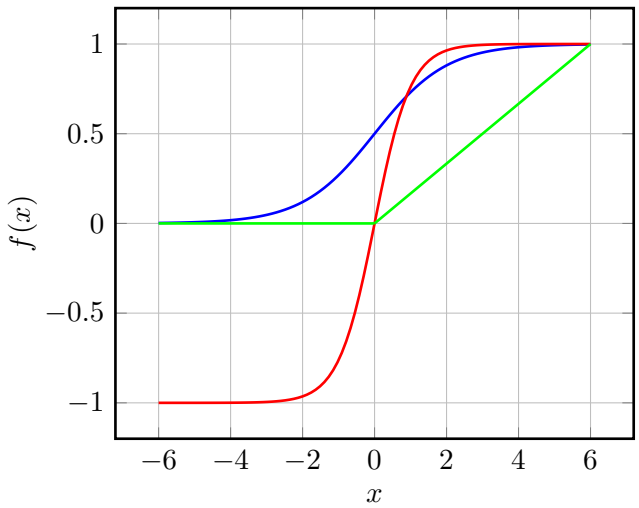


Figure 2.9: Plot of activate functions, $sigmoid(x)$ in blue, $tanh(x)$ in red and $relu(x)$ in green.

2.2.1.6 Training Neural Network: Back Propagation

Back Propagation (BP) refers to the backward propagation of errors. It was originally conceived in the 1970s by Werbos and became a well-known methodology after the study by Rumelhart et al. (1988).

The algorithm first calculates the error according to the loss function described in Section 2.2.1.5. In most cases nowadays, unlike simple 3-layer MLP ANNs, it can be a very complex function especially when multiple layers, such as a CNN layer or a softmax layer are involved. Generally, an ANN can be written as a function of stacked hidden functions. See Equation 2.11 for an example of an MLP network.

$$\boldsymbol{y} = f(\boldsymbol{x}) = h^{(l)}(W^{(l)}h^{(l-1)}(W^{(l-1)}h^{(l-2)}(\cdots(h^{(1)}(\boldsymbol{x})))))) \tag{2.11}$$

where $\boldsymbol{x}$, $\boldsymbol{y}$ is a training sample, $h^{(l)}$ is the hidden function of layer l, and $\boldsymbol{W}^{(l)}$ is the weight matrix for layer l. The loss function ℓ is defined as:

$$\ell = loss(\boldsymbol{y}_o, \boldsymbol{y}) \tag{2.12}$$

To obtain the best weights, an optimising process, usually a *gradient descent* algorithm, is performed. The process has two steps. First, the input vector $\boldsymbol{x}$ is *forward*ed and an output vector $\boldsymbol{y}_o$ is generated accordingly. The second step is to back propagate the error, also called *backward*, which reversely determines how much error is contributed by each weight.

Traditionally, *backpropagation* can be achieved by applying a *chain rule* while solving the derivatives layer by layer. From another point of view, the expression of the loss function, with weights as its independent variables, can be seen as a super surface that contains a lot of hills and valleys, and one combination of weight values can be seen as a point on that multi-dimensional surface. The task of backward is to find a minimal value on the surface when the

exit criterion is satisfied. With the help of auto-gradient solvers like tensorflow (Abadi et al., 2015), theano (Theano Development Team, 2016), caffe (Jia et al., 2014) etc, nowadays it is possible to directly obtain the gradients, even though the network has millions of weights to train.

The gradient descent can be described as a "down-hill" algorithm. For each step, it finds a direction by solving partial derivatives for each target weight, updating the target weights with a specific step size, which is called the *learning rate*, so that a lower cost can be achieved according to the loss function. In order to obtain the "down-hill" direction for each target weight, its partial derivatives are calculated according to the loss function. Algorithm 1 is the gradient descent algorithm.

Algorithm 1 Gradient Descent

Require: loss function ℓ, target parameters to optimise $\mathbb{W}$, training samples $\boldsymbol{X}$ and $\boldsymbol{Y}$, learning rate η

initialise $\mathbb{W}$
procedure GRADIENT DESCENT($\boldsymbol{X}$, $\boldsymbol{Y}$, η)
    initialise $\mathbb{W}$
    while exit criterion not satisfied **do**
        pick a pair of $\boldsymbol{x}$ and $\boldsymbol{y}$ from $\boldsymbol{X}$ and $\boldsymbol{Y}$
        $\boldsymbol{y}_o \leftarrow f(\boldsymbol{x})$
        $g \leftarrow \nabla \ell(\mathbb{W})$
        update target parameters: $\mathbb{W} \leftarrow \mathbb{W} + \eta \cdot g$
    end while
end procedure

2.3 Natural Language Processing

2.3.1 Preprocessing

2.3.1.1 Tokenise

Tokenisation refers to the task of splitting a sentence into chunks, each chunk is called a token, often this is a word, a number or punctuation mark. For example, consider the following title of a news article:

Brexit: EU workers 'adding £7.3bn to Scottish economy'

Proper tokenisation will split it into the following tokens:

Brexit / : / EU / workers / ‘ / adding / £7.3bn / to / Scottish / economy / ’

Most tokens can be obtained by splitting a sentence using white space or punctuation. However, it is still challenging to tokenise a sentence containing phone numbers (e.g. +44 01904 320 000), dates (Mar 20, 1987), special nouns (e.g. C++, X-10), and borrowed foreign words (l'enclume) etc.

2.3.1.2 Stemming and Lemmatisation

In English, a verb is used in different forms for grammatical reasons, for example *write*, *writes*, *wrote* and *written*. There are also words which share the same meaning and stem but in different parts of speech, e.g. *effect*, *effective* and *effectiveness*. In most cases, morphological variants of words have similar semantic interpretations and can be considered to be equivalent for the purpose of Information Retrieval. *Stemming* refers to converting a word to its stem or root form following a group of rules (e.g. Porter Stemmer (Porter, 1980)). *Lemmatization* is a similar task to *stemming*, the difference is that it takes morphology and vocabulary into account rather than directly removing or converting words into stems.

2.3.1.3 Part-of-Speech Tagging

Part-of-Speech (PoS) tagging refers to assigning tags to a word according to its role in a sentence, based on context. PoS tags provide information about the semantic content of words. Noun, verb, article, adjective, preposition, pronoun, adverb, conjunction and interjection are the most commonly used PoS tags. The PoS tags depend on specific tasks, sometimes only 3-4 PoS tags are used, and in some cases PoS tags are divided into subcategories. Consider the following sentence:

> They refuse to permit us to obtain the refuse permit.

after PoS tagging:

> They/PRP refuse/VBP to/TO permit/VB us/PRP to/TO obtain/VB the/DT refuse/NN permit/NN ./.

Note there are two words 'permit' with different PoS tags, the first is VB (verb in base form), and the second is NN (noun).

Various PoS taggers can be implemented using machine learning techniques like HMM, Cyclic Dependency Networks, Concept Learning or Deep Learning.

2.3.2 Document Representations

2.3.2.1 Bag-of-Words

As its name suggests, the bag-of-words (BoW) model simply treats texts as a bag with words without considering ordering between words or sentences. For example, the sentence

> Brexit: EU workers 'adding £7.3bn to Scottish economy'

can be represented as

```
[brexit, eu, workers, adding, £7.3bn, to, scottish, economy]
```

If using a unigram model, each element in the bag is usually a token.

BoW models are commonly used to generate frequency or occurrence-based features. A lot of higher level models can be built on the basis of a BoW model. For example, one of the most widely used models, *Term-Frequency and Inverted Document Frequency* (TFIDF), is an additional weighting scheme that is often used to reduce the importance of words that appear across most documents, and highlight the ones that are characteristic of a small subset of documents (Salton and Buckley, 1988). The relative frequency of word w in document d

is weighted according to Equation 2.13. This reduces the perceived importance of word w in document d to zero if the word appears in all documents, and increases it gradually as the number of documents containing w decreases (Sedding and Kazakov, 2004).

$$tfidf_{d,w} = \frac{freq_{w,d}}{|d|} \cdot log \frac{|\mathcal{D}|}{|\{d \in \mathcal{D} : w \in d\}|} \tag{2.13}$$

where

$|\cdot|$ is the size of a set.

$freq_{w,d}$ is the number of occurrences of word w in document d.

2.3.2.2 Distributional Representation and Embedding Techniques

The aim of distributional representation is to represent words in a vector with a fixed number of dimensions that is much less than the vocabulary size. The Neural Network Language Model (NNLM) proposed in Bengio et al. (2003) can be seen as one of the most important in this field. An NNLM has three layers; the first layer is to map n words to n vectors (aka embeddings or Word2Vec vectors in later texts). The second layer is a hidden layer that transforms input vectors into the third layer by applying an activate function ($tanh$). The task of the NNLM is to predict the next word given the previous n words, so the last layer is a $softmax$ layer that outputs the probabilities of each word occurring.

Word2Vec proposed by Mikolov et al. (2013) can be seen as a intermediate product when an NNLM is obtained. There are two approaches To obtaining Word2Vec vectors: CBOW and SkipGram. The latter approach, the Word2Vec model can be seen as an one-layer neural network, where the inputs are vectors of words that surround the target word, called the context window, and the output is the vector of the word which is surrounded.

2.4 Statistical Measures of Correlation

2.4.1 Dynamic Time Warping

Dynamic Time Warping (DTW) is used for measuring the dissimilarity between two time series and is widely used in speech recognition. One feature of this algorithm is that it does not require two series to have the same length. For example, the word 'the' may be spoken over different lengths of time, but the patterns of the different pronunciations will appear to be very similar visually. The idea of the DTW algorithm is to find the best alignment (warping path) between two time series. This feature overcomes the weakness of sensitivity to distortion in the time axis compared to the Euclidean distance (Keogh and Ratanamahatana, 2005).

Formally, consider two time series X and Y:

$$X = x_1, x_2, \cdots, x_i, \cdots, x_m$$

$$Y = y_1, y_2, \cdots, y_i, \cdots, y_n$$

The potential alignments between X and Y can be represented in an $N \times M$ matrix where $m_{i,j}$ corresponds to the distance between x_i and y_j. The DTW distance can be solved using dynamic programming to evaluate the following recurrence:

$$d(i,j) = m_{i,j} + min\{d(i-1,j-1), d(i-1,j), d(i,j-1)\} \tag{2.14}$$

where

$$DTW(X,Y) = d(m,n) \tag{2.15}$$

2.4.2 Word Mover's Distance

Similarity measuring (or distance/dissimilarity measuring) is an essential task of document retrieval systems. The Bag-of-Words (BoW) and TFIDF are the most commonly used repre-

sentations when a similarity is needed. However, there are drawbacks when using them: 1) the representations used in BoW and TFIDF have a huge number of dimensions and the obtained vectors are sparse. This usually causes calculation difficulties. 2) The distance functions, commonly Cosine distance or Euclidean distance, do not capture similarity between words with a close meaning.

The Word Mover's Distance(WMD) proposed in (Kusner et al., 2015) shares the same idea and part of the calculation of the EMD (Rubner et al., 1998), and also DTW to some extent. The EMD is a measure of the distance between two distributions within the same region, given:

$$\mathbb{P} = \{(p_1, w_{p_1}), \cdots, (p_m, w_{p_m})\}$$

$$\mathbb{Q} = \{(q_1, w_{q_1}), \cdots, (p_m, w_{q_n})\}$$

$$\mathbf{D} = [d_{ij}]$$

where $\mathbb{P}$ and $\mathbb{Q}$ are two signatures with m and n clusters, p_i and q_j are cluster representatives, w is the weight of the corresponding cluster, $\mathbf{D}$ is the ground distance matrix, and d_{ij} is the distance between p_i and q_j. The EMD distance is the best solution to the transportation problem that transports from $\mathbb{P}$ to $\mathbb{Q}$ with least cost. Formally, the EMD minimises the overall cost

$$Work(\mathbb{P}, \mathbb{Q}, F) = Minimise(\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}) \tag{2.16}$$

subject to

$$
\begin{aligned}
f_{ij} \geqslant 0, 1 \leqslant i \leqslant m; 1 \leqslant j \leqslant n \\
\sum_{i=1}^{n} f_{ij} \leqslant w_{p_i}, 1 \leqslant i \leqslant m \\
\sum_{i=1}^{m} f_{ij} \leqslant w_{q_j}, 1 \leqslant j \leqslant n \\
\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = min\{\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j}\}
\end{aligned}
$$

and the EMD can be obtained using:

$$
EMD(\mathbb{P}, \mathbb{Q}) = \frac{Work(\mathbb{P}, \mathbb{Q}, F)}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \tag{2.17}
$$

In a WMD scenario, $\mathbb{P}$ and $\mathbb{Q}$ are two documents in which words are represented by vectors, in other words, p_i and q_j will be word embeddings if documents are represented by a Word2Vec model. w_{p_i} is the normalised word frequency of i-th word in $\mathbb{P}$. The distance d_{ij}between p_i and q_j is provided by their Euclidean distance. f_{ij} refers to how much earth moves from i-th word to j-th word and $\sum_j f_{ij} = w_{p_i}$. In our research, since the Mantel test (introduced in section 4.4) requires distance matrices to be symmetrical, **Related Word Moving Distance** proposed in (Kusner et al., 2015) is used to obtain WMD where

$$
f_{ij} = \begin{cases} w_{p_i}, & \text{if } j = argmin_j w_{p_i} \\ 0, & \text{otherwise} \end{cases} \tag{2.18}
$$

This is equivalent to find pairs of words from two documents such that each pair of words has the closest distance to each other. The final WMD is obtained by

$$
Max(EMD(\mathbb{P}, \mathbb{Q}), EMD(\mathbb{Q}, \mathbb{P})) \tag{2.19}
$$

The WMD has several appealing properties: it is hyper-parameter free, highly interpretable, especially when incorporated with the Word2Vec model, and its performance outperforms most of the alternative distance metrics (Kusner et al., 2015).

2.5 Related Work

There has been considerable research in recent years in the field of textual based financial forecasting. The research can be categorised into groups based on methodologies. In order to compare different approaches, typical studies were picked and summarised by answering the following questions:

- Aim: What does the research forecast?
- Methodology: What is the methodology used for forecasting?
- Data: What textual data and numerical data is used?
- Result: How is the forecasting performance?
- Comment: What are the merits and demerits of the research?

AZFinText implemented by by Schumaker and Chen (2009) was developed at the University of Arizona. This study aims to forecast stock prices 20 minutes ahead of time.

The system consists of four components: Textual Analysis, Stock Quotation Analysis, Machine Learning Algorithm and Error Analysis, as shown in Figure 2.10. In the training phase of SVR, two prices are used. One is the estimated price 20 minutes ahead of article release, denoted as $\hat{p}_{t+20}$, which is calculated using a regression of a 1-hour window from article release. The other price is the real price p_{t+20}. Text features with $\hat{p}_{t+20}$ are used as training data with p_{t+20} as the target value.

In order to compare performances of different text representations, in the Textual Analysis component, AzTek system (McDonald et al., 2005), a word level tagging system, is used to

implement Bag-of-Words, and extract named entities and noun phrases from news articles. For each stock, quotations are gathered at the minute level, then a linear regression analysis is performed using data 20 minutes prior to an article release to forecast the 20 minutes ahead stock price. Support Vector Regression is used as the machine learning algorithm.

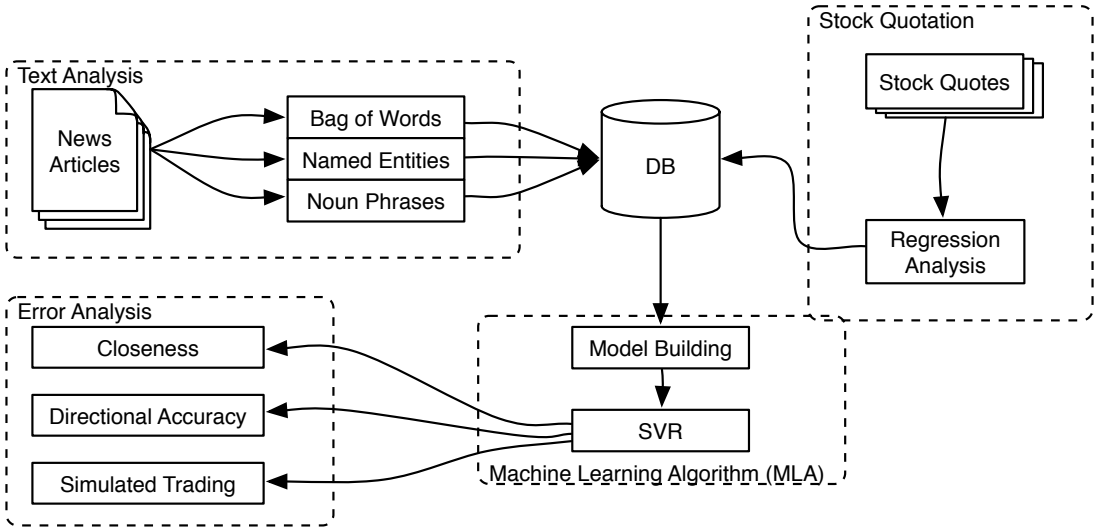


Figure 2.10: The structure of AZFinText.

Mittermayer and Knolmayer (2006) developed a trading system called NewsCATS (News Categorization and Trading System). This system aims at short-term stock price forecasting. There are three components in NewsCATS: pre-processing, categorisation and trading (see Figure 2.11). The pre-processing component is in charge of 1) using the Bag-of-Words model to extract features from press releases, and 2) capturing features based on a handcrafted thesaurus which includes signal words, phrases or logic tuples of words and phrases such as 'sell NEAR financial crisis'. The categorisation component used a trained SVM classifier built from the SVM^{light} package (Joachims, 1999) to place documents into 3 categories, good, bad and neutral, which is a very commonly approach in text-based stock market forecasting. Finally, the trading component gives a trade recommendation of a corresponding security.

A simulation was made based on 15-second intervals of S&P500 stock transaction data and *US PRNewswire* archived press releases of companies from 2002-04-01 to 2002-12-31. The

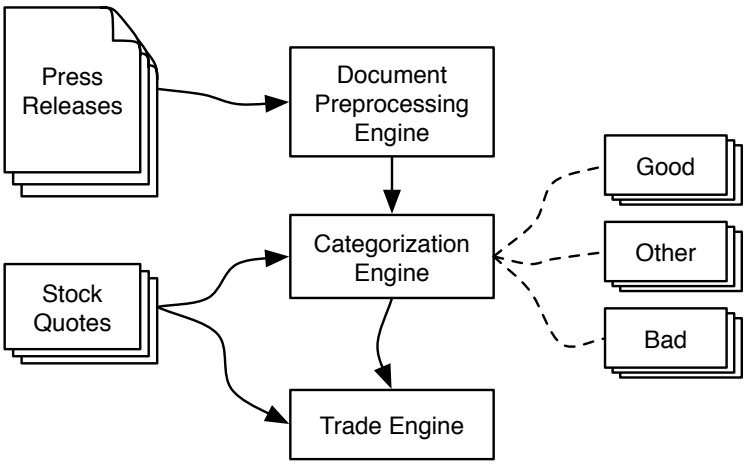


Figure 2.11: The structure of NewsCATS.

simulation result shows 0.23% profit per roundtrip. The authors also analysed robustness by tuning parameters to make a comparison of different feature selection functions, feature sets, document representations and classifiers. The analysis indicated that NewsCATS can perform robustly with the given dataset.

This research is actually a text categorization system rather than a forecast-based trading system. Textual data only provides limited information by giving a good, bad or neutral signal to aid in making a trading decision.

Wang et al. (2012) proposed a text mining approach which combines both ARIMA and SVR for forecasting. This study makes daily stock price forecasting for 3 Chinese and 3 US company stocks according to *Best Investor Relationship Award* magazine.

The forecasting problem is split into two parts: linear forecasting and nonlinear forecasting. Numerical time series data is treated as the source data for the linear forecasting part, and an ARIMA model was built to make the forecast. This is a common approach in stock price prediction. The structure of this forecasting system is shown in Figure 2.12.

Textual data is represented using a Bag-of-Words model for nonlinear forecasting, labelled using +1 shift time series data, and SVR is used as the classifier. Part of the quarterly and annual

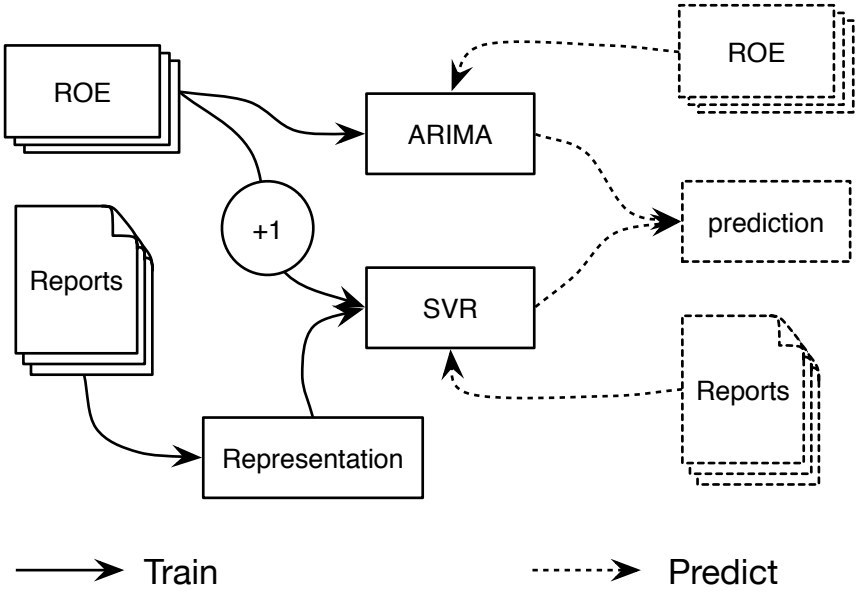


Figure 2.12: The structure of the SVR + ARIMA forecasting model.

reports from *Management Analysis and Discussion* are used as textual data, while corresponding quarterly ROE (Return On Equity) are selected for the time series data.

No clear metrics from a financial point of view were shown in the results. However, such a hybrid model shows significant improvement technically compared with ARIMA alone or using SVR as the only model.

Zhai et al. (2007) proposed a forecasting system which combines company specific news and general market news with price history to forecast daily stock prices. In this research, two groups of news releases are used as textual data. One was directly related to the stock and the other was related to the general market. A Vector Space Model is adopted to represent news releases, but instead of using the original words, the authors use WordNet to replace the original words with higher-level concept words. Words with a top 30 $tfidf$ weight are selected as feature words to code news releases, and an SVM classifier is used to categorise news releases into either a good or bad basket.

BHP Billiton Ltd. (BHP.AX) was selected as the experiment stock. This is a relatively

large volume stock on the Australian stock market in the metal sector. Textual data is from news articles published from 2005-3-1 to 2006-5-31, in the Australian Financial Review, a business, finance and investment newspaper. Stock quotes in the same period are used as numerical data. Results show that an accuracy of 70.1% was achieved compared with numerical, which was only 58.8%. Although some details are missing, the idea of using WordNet to help textual forecasting is innovative and worth investigating in future.

An Event-Driven Forecasting System by Fung et al. (2005) presents a system to predict stock trends using news articles. The system is event-driven instead of using a fixed time interval.

The system structure is shown in Figure 2.13. Given a stock, it can be summarised as follows: In the training step, for the numerical part, a t-, the test based piece-wise segmentation algorithm is used to discover a trend in stock movements. The trends are then clustered and labelled. For the textual part, news articles are clustered and aligned to clustered trends. According to a newly proposed algorithm, news articles not in the trend will be filtered out; the rest of the news articles are represented using a tf-idf variation. An SVM classifier is used to label documents with either *rise* or *drop* and another classifier determines whether or not the event is triggered.

In the experiment, three stocks from the Hong Kong market and data from five consecutive months, from 2003-1-20 to 2003-6-20 were used, comprising 350,000 news articles extracted from Reuters Market 3000 Extra. Since this system only forecasts stock trends, unlike other forecasting systems, no specific range of time is given, so hit rate is used as a metric instead of accuracy. The results show that this system can achieve up to 61.5%-65.4% hit rate if predicting three to five days ahead.

The idea of segmenting curves into larger trends was pioneered by Pavlidis and Horowitz (1974), and Lavrenko et al. (2000) incorporated this technique for stock forecasting. Seg-

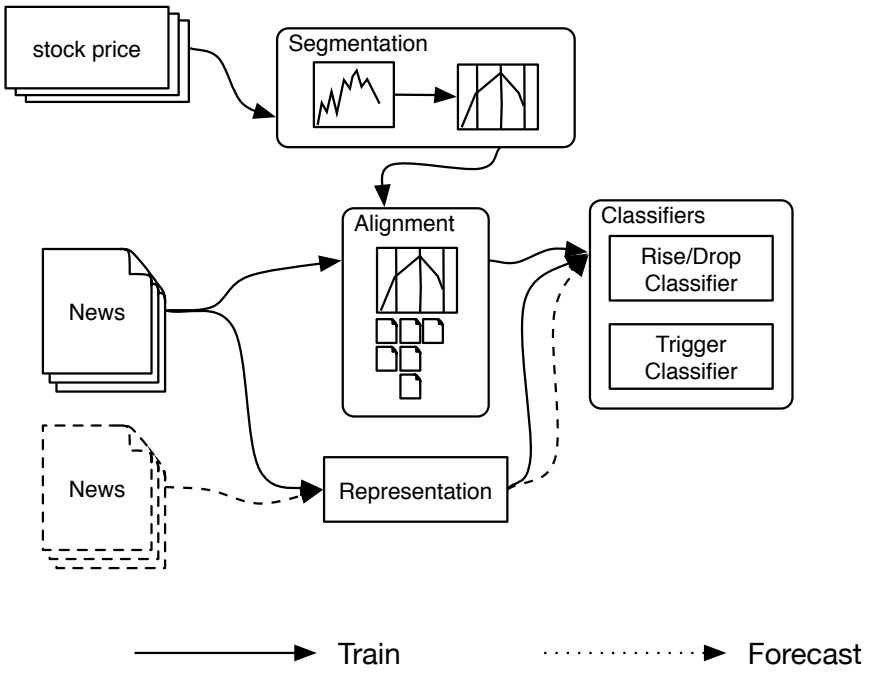


Figure 2.13: The structure of the event-driven forecasting model by Fung et al. (2005).

mentation is very commonly used by human analysts. Such techniques can eliminate noisy volatilities and reveal the basic movements of a stock. Moreover, event-driven forecasting is also worth discussing as an open problem. Kogan et al. (2009) aim to forecast the volatility of stock returns, which is an empirical measure of financial risk.

The textual data were extracted from Section 7 of the 10-K reports of listed stocks. This section is called "Management's discussion and analysis of financial conditions and results of operations" and contains the most predictive information. Numerical data is stock volatility calculated from stock prices. Each document is represented by three representations: tf, $tfidf$ and $log1p$ (logarithm of 1 plus the frequency of each word in a document). The idea is to model the relationship between volatility and the weights of documents, where SVR is used as the learning algorithm to fit proper weights to the model. 54,379 10-K reports between 1996-2006 of 10,492 companies are used as textual data. Stock prices from the same period are calculated for volatilities as numerical data. The authors use MSE to evaluate the volatility

forecasting. Although the results are not comparable with others, it illustrates well that by introducing textual data, numerical forecasting can be improved.

With the successful application of Deep Learning, in particular in the areas of computer vision and natural language processing, research has been done using deep learning to boost financial forecasting, applied in either time series forecasting or textual feature representation, for example Word2Vec.

Si et al. (2014) developed a novel financial forecasting approach that utilises cash-tagged '$' stock symbol in tweets. The approach first detects co-occurrence of pairs of stocks and builds a semantic network (Semantic Stock Network)in which nodes and edges are assigned with a topic respectively by a labelled topic model. The authors illustrate that social sentiment about topics and stock relationship contains predictive power.

The experiment contains a very limited number of stocks and gives bi-directional accuracy on a daily time interval, in which the best accuracy reported is up to 0.7 and most accuracies vary around 0.55. Overall the authors present a study that implies that pair-wise relations between stocks can be utilised for extracting predictive information.

Ding et al. (2015) propose a deep learning method to make stock predictions using online news. The authors first extract events from news texts using Open IE, an open source information retrieval toolkit. These are represented as tuples in the form of $E = (O_1, P, O_2, T)$, where P is the action, O_1 is the actor, O_2 is the object which received the action, and T refers to the time-stamp. Those extracted events are represented as event embeddings using a pre-trained Neural Tensor Network, which takes word embeddings of the event tuples as inputs and outputs event embeddings. The prediction model is a binary classification model based on a hybrid network with convolutional layers to decide if input events are positive or negative. The input of the network has three components due to the consideration that the effect of events may vary over time. Therefore, there is a 30-day long-term window, a 7-day mid-term

window and a dense layer to capture daily events. Since there can be more than one event in one day, events in one day are averaged as one day's input.

The experiment covers financial news from Reuters and Bloomberg from October 2006 to November 2013. The best directional accuracy of the research reaches 65.08% on the S&P 500 index. It is also noticeable that embeddings based on the word level, instead of the extracted event, can reach 61.73%.

This research shows the effectiveness of applying deep learning techniques to financial forecasting. With embedding techniques and a convolutional layer, the feature extraction step is no longer the most difficult procedure when extracting useful information. However, training a deep learning network can be hard work, and it is usually impossible to obtain the same good results even using the same hyper-parameters. This disadvantage may lead to unstable performance in the financial forecasting pipeline.

2.5.1 Summary

So far, we have reviewed representative approaches above, other existing researches are similar to one or some of these typical researches. A general textual financial forecasting system is described in Figure 2.14.

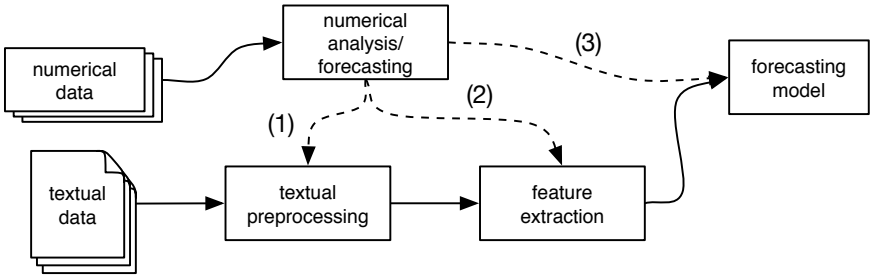


Figure 2.14: The structure of a generalised textual financial forecasting system.

- Numerical Data: one of two raw inputs of a forecasting system. It usually consists of stock quotes, index average values or other financial time series data related to the forecasting

problem.

- Textual Data: another raw input, consists of natural language texts that are directly or indirectly related to the forecasting problem.
- Numerical Analysis/Forecasting: different analysis techniques can be used in this component. For example in Fung et al. (2005), segmentation is applied to discover trends, and in Wang et al. (2012) ARIMA is used for numerical forecasting.
- Textual Pre-processing: data cleaning and NLP pre-processing tasks are performed, such as tokenisation, PoS tagging and stemming.
- Feature Extraction: different models can be applied for textual document representation. The vector space model is the most commonly used. Other techniques like sentiment analysis can also be applied.
- Forecasting Model: machine learning techniques can be used in this component. SVM for instance, is the most commonly seen classifier.
- Connections: sometimes numerical data are part of the textual data. Connection (1) refers to document filtering or other operations, for example, in Fung et al. (2005), less predicable documents are removed, and in Mittermayer and Knolmayer (2006), numerical data are used for aligning and categorising documents into three groups: good, bad and neutral. For connection (2), most studies do not have it, which means that numerical data is directly incorporated into the feature selection stage as in Fung et al. (2005). Connection (3) means that numerical data is directly used as an input in the forecasting model.

Chapter 3

Data Collection and Exploration

3.1 Introduction

Most research in the field of NLP uses standard and publicly available datasets, e.g., the widely used classic text collection Reuters-21578, for the purpose of making comparisons with other research using the same standard. However, for financial forecasting, the scope cannot be defined as clearly as typical NLP tasks, such as Named Entity Extraction, Question Answering or Text Classification. Almost all individual research projects use their own dataset, which largely depends on their own favoured approach. Intra-day forecasting for instance, requires lower time-framed data than longer-term forecasting. Therefore, researchers tend to use tweets rather than news articles. US located researchers use news from US markets, while European researchers collect news from both the US and Europe. Moreover, datasets used in the existing literature are barely available publicly, sometimes for legal reasons. Therefore, a standard specialised dataset for research into financial forecasting does not exist. A data acquisition system is necessary to collect both textual data and numerical data which can be used in this research.

3.2 Data Sources

There are a good amount of financial text sources available nowadays. Those sources of text can be categorised into seven groups and measured by five metrics:

Comprehensiveness measures how much a text source covers an event and the information it provides. If a text source requires a lot of context to understand, its comprehensiveness is considered to be low;

Expertise measures how professional text sources are. A professional text source is considered able to provide better analysis and a deeper view of an event.

Immediacy measures how fast text sources can respond to an event.

Accuracy measures how correctly text sources report an event.

Availability measures how easily text sources can be acquired.

Table 3.1 shows the seven text sources and their performance based on these metrics.

Table 3.1: Different textual data sources for finance. There are five metrics to measure: Co* = Comprehensiveness, Ex* = Expertise, Im* = Immediacy, Ac* = Accuracy, Av* = Availability. H = High, M = Medium, L = Low.

Source	Co*	Ex*	Im*	Ac*	Av*
Company Reports	H	H	L	H	L
Analyst Reports	H	H	L	M	L
News Wire Services	M	M	H	M	H
Forums	L	L	M	L	M
Blogs	L	L	L	L	H
Social Networks	L	L	H	L	H

Almost all researchers use news articles from news wire services or Twitter feeds. Very few use company reports or analyst reports; although they are much more comprehensive, with higher expertise and accuracy, their low immediacy and availability sets a great barrier to short-term or mid-term forecasts. Compared to news wire services, forums, blogs and social networks contain massive noise; people can publish any textual data, and they do not have to

care about the content in the way that a news article reporter does. For example, in a forum post or tweet, jargon and informal abbreviations are very often used, which require great effort to properly pre-process.

Taking in to account the considerations above, an online news wire service was chosen as the text data source.

3.3 Parsing Online News

Yahoo provides an RSS[1] news service to the public. The advantage of the RSS news service is that news is divided into different channels. For example, news related to Apple will appear in its RSS feed, while Google's RSS feed will update with all the news about Google. Some news may be shared over several companies' RSS feeds.

In order to continuously fetch all the news articles from Yahoo Finance RSS feeds for the selected stocks, an automatic extraction system was implemented and configured to check for new articles every ten minutes. The structure of the news extraction system is shown in Figure 3.1.

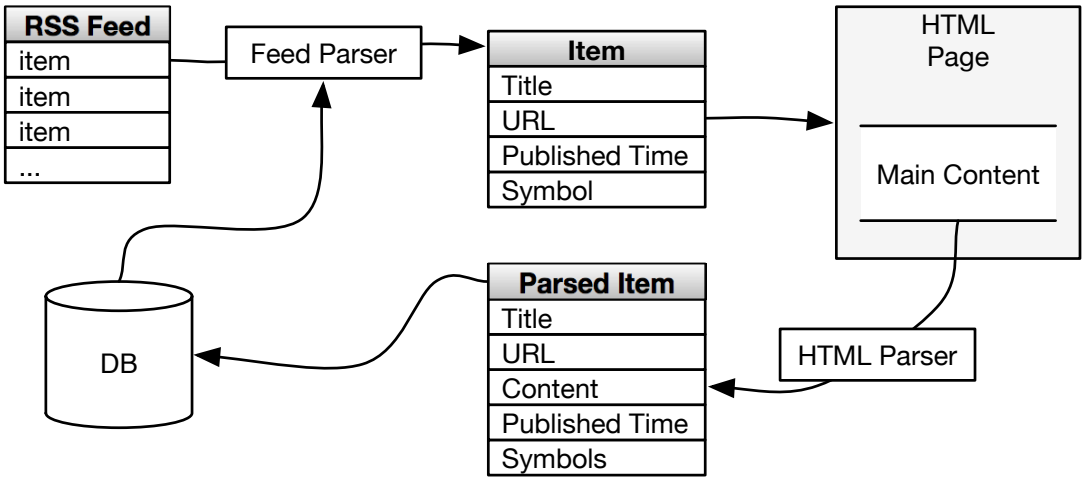


Figure 3.1: Structure of Yahoo Finance News RSS extraction system

[1]RSS refers to RDF Site Summary or Really Simple Syndication, which uses standard web feed formats to publish information like news or blogs that are frequently updated.

In Figure 3.1 the components are as follows:

RSS Feed Each stock has its own RSS feed that contains no more than 20 items. It works like a stack; when new items are published, the same number of old items will be removed from the feed. Also, old items have a lifespan. Even if there are no new items pushed in, expired items will be removed. Two feeds extracted at a different time may contain duplicate items.

Feed Parser The Parse RSS feed extracts all items and filters out duplicate items.

Item Items extracted from the RSS feed have some basic information: title, URL of the news article web page, published time and the stock symbol. Note that one item may appear in different feeds of stocks at different times.

Main Content Parser Parses main content from the raw web page of each news item. Since all news items are from portal websites, although those sites have different page frameworks, the basic patterns, with main contents located in an HTML DOM tree, are very similar. Once all the noisy element containers such as navigations, advertisements, sidebars and footers are filtered out, in the rest of the containers, one has the longest text, which is the most likely to contain the main content.

Parsed Item Main content is added into each item and wrapped as an object to be loaded into the database.

All news articles are pre-processed using NLP tasks after being extracted from the database. The procedure is shown in Figure 3.2. The implementation mainly takes advantage of the Python Natural Language Toolkit package (Bird, 2006) which is a platform on which to write Python programs to work with natural language data.

In Figure 3.1 the components are as follows:

- Tokenisation: first use a rule-based sentence tokeniser to split the main content into sentences, and then a regular expression based tokeniser turns sentences into tokens.
- PoS tagging: a Maximum Entropy tagger trained using the Treebank corpus is used as a PoS tagger to assign PoS tags.
- Lemmatisation: After each word is assigned a PoS tag, a Word-Net based lemmatiser will transform words tagged as a noun, adjective, adverb or verb to their base form.
- Named Entity Recognition: A Stanford NER tagger (Dingare et al., 2005) was initially used to recognise named entities, but the performance by random check is not very satisfying, so currently it is not enabled.
- Word2Vec : look up using pre-trained embeddings from Komninos and Manandhar (2016).

3.4 Explorations of Textual Data

As the dataset is always growing, a snapshot of the dataset was taken on 2013-5-2. Data was gathered from 2013-1-24 to 2013-5-2, a total of 91 days. Figure 3.3 shows the logarithm of the total number of news articles extracted for each stock, ranked from high to low. It shows that stocks obey a rule similar to Zipf's law: the logarithm number of news articles and their rankings are linearly correlated.

Figures 3.4 and 3.5 show a box-plot of 24-hour and weekday quantity distributions of news articles for Apple Inc. (NASDAQ: AAPL) and Google Inc. (NASDAQ: GOOG) published in Yahoo Finance RSS feeds. The 24-hour distribution shows that news article publishing follows common human work and rest times, and this partly overlaps with the market schedule. The hourly news publishing rate gradually goes up after the market opens and continues until after market hours end at 8pm. The weekly distribution shows that the amount

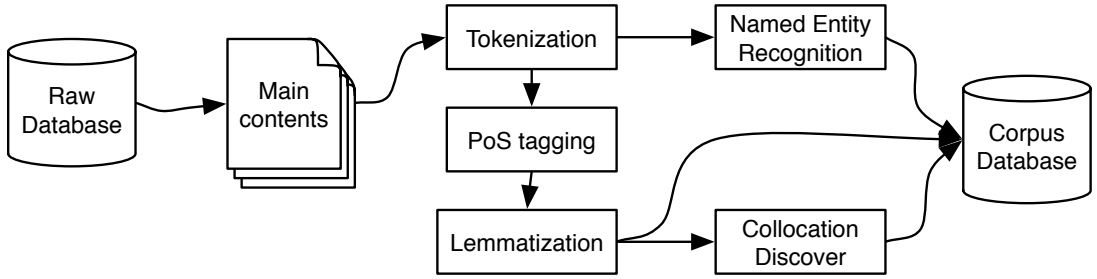


Figure 3.2: Structure of pre-processing system

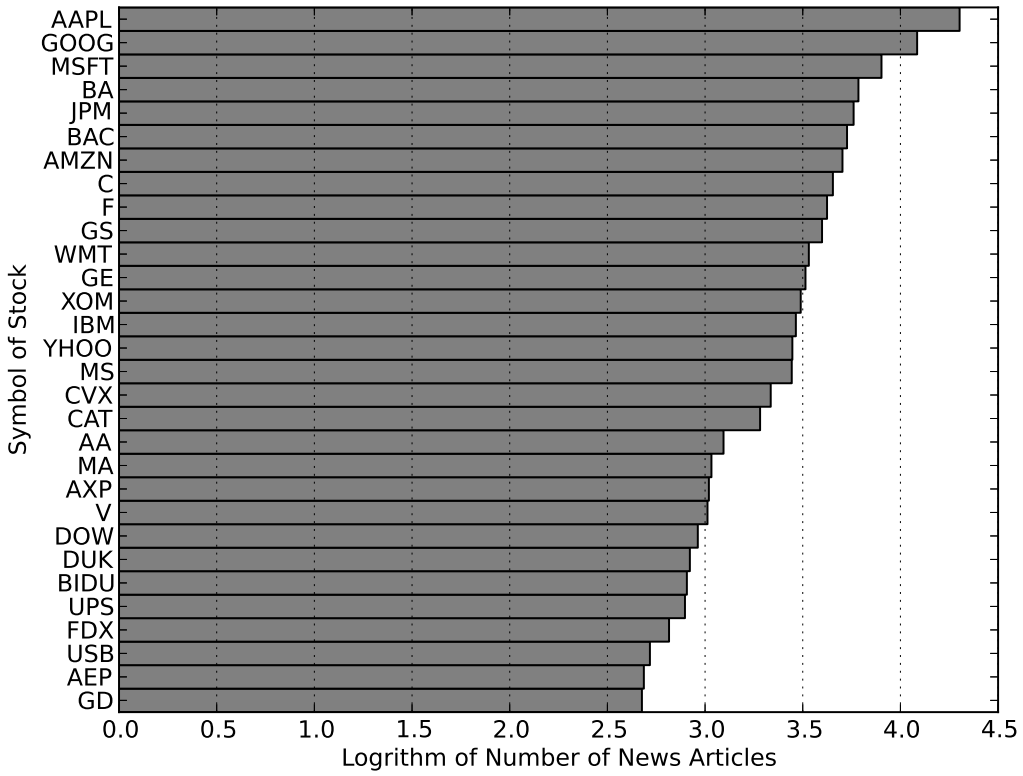
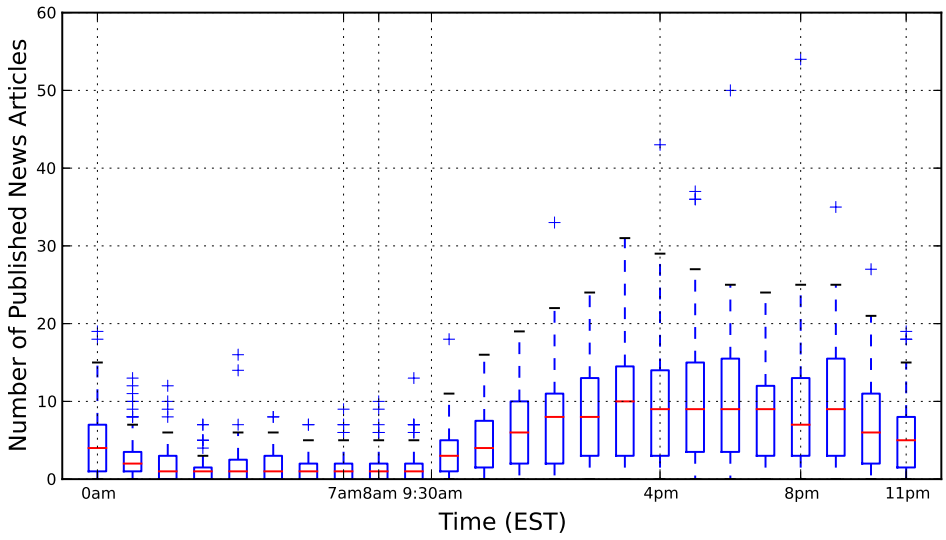
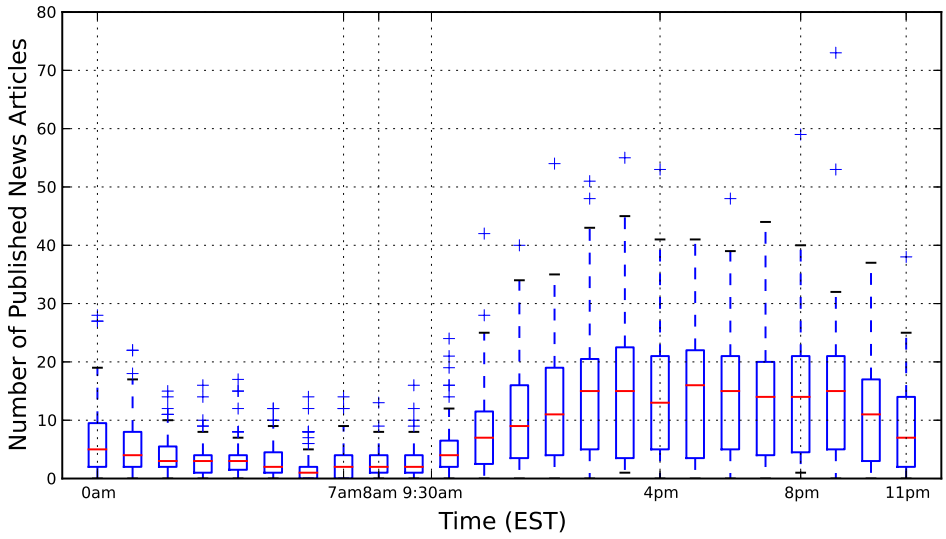


Figure 3.3: Logarithms of total number of news articles for stocks, ranked from high to low.

of news published at the weekend is significantly lower than on weekdays and mid-week days have a higher number of news articles.

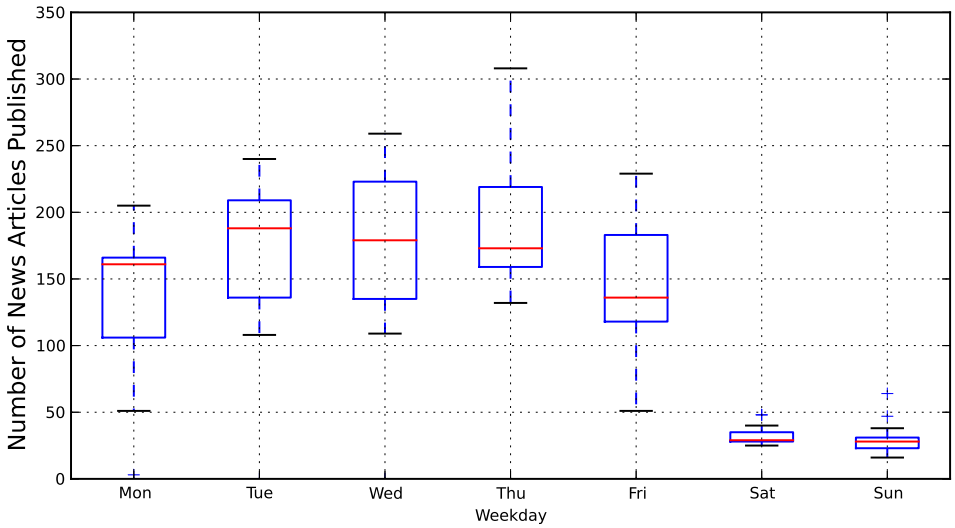


(a) GOOG

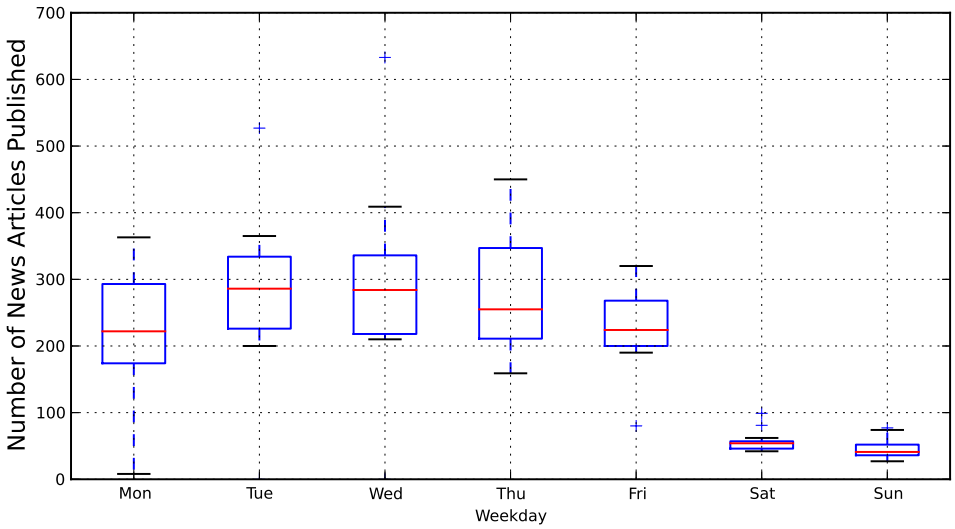


(b) AAPL

Figure 3.4: 24 hour distributions of the number of financial news articles published in Yahoo Finance RSS feeds, for AAPL and GOOG, two NASDAQ listed stocks (data from 2013-1-28 to 2013-4-28). Market trading time for NASDAQ is between 9:30am & 4:00pm EST.



(a) GOOG



(b) AAPL

Figure 3.5: Weekly distributions of the number of financial news articles published in Yahoo Finance RSS feeds, for AAPL and GOOG, two NASDAQ listed stocks (data from 2013-1-28 to 2013-4-28). Market trading time for NASDAQ is between 9:30am & 4:00 pm EST.

Chapter 4

Quantifying the Relationship between Financial News and Time Series

4.1 Introduction

An essential precept of a financial market is that asset prices break their status or pattern because of the impact of unforeseen extra information, which is usually delivered in the form of publicly available news. There are numerous studies (Boudoukh et al., 2013; Ding et al., 2015; Foucault et al., 2012; Gidofalvi and Elkan, 2001, *inter alia*), on exploiting textual data to extract useful information to help with financial decisions, such as asset selection, position optimisation or risk estimation. Most work follows the approach of extracting the tone of a piece of text (e.g. positive, negative or neutral), to provide additional assistance with financial forecasting or analysis. For example, an early study conducted by Tetlock (2007) used General Inquirer, an NLP analysis tool with a psycho-social dictionary (Harvard IV-4), to count words which fall into different categories.

Boudoukh et al. (2013) noticed the problem that publicly available financial news may not be helpful in providing relevant information. However, accurately classifying news articles as relevant to stocks is challenging work and requires a huge amount of effort from human financial experts. On the other hand, the dynamics of financial markets are constantly changing.

The reactions of markets to the same news varies and depend on many factors, and the complexity of this chaos cannot be well modelled. The aim of the present work is to construct a framework that passively quantifies the relationship between financial news and market trading data. The results are encouraging and show that our approach outperforms state-of-the-art industry trading strategies. We believe that we have successfully designed a method to quantify the relationship between financial news and time series. This novel framework provides a passive indicator, meaning that instability in the stock market in reaction to news items can be greatly eliminated. Such a tool could be used to aid portfolio selection. Throughout this chapter, the terms 'time series' and 'stock trading data' are used interchangeably.

The chapter is organised as follows: Section 4.2 briefly overviews the proposed framework. Section 4.3 gives details of how the distance aggregation methods for the news and time series were obtained, which is the basis of the proposed framework. Section 4.4 outlines the methodology for quantifying the relationship between news and time series based on the Mantel test. In Section 4.5 the data and experiment are presented. Section 4.6 presents the results obtained from the proposed framework. Section 4.5.3 provides an evaluation and comparison between the proposed framework and common baselines. Conclusions are drawn in section 4.7.

4.2 Overview of Framework

Textual information is supposed to be the factor that drives financial markets for a certain period. Previous studies often tried to assign or find explicit links between textual information and time series, e.g. (Mittermayer and Knolmayer, 2006; Robertson et al., 2007; Schumaker and Chen, 2009). Instead of actively modelling the relationship between news and time series, we are interested in the relationship between the news and time series in terms of passively observing and measuring. The framework we propose contains several components (see Figure 4.2 for details). Data Acquisition is responsible for collecting textual data from news and

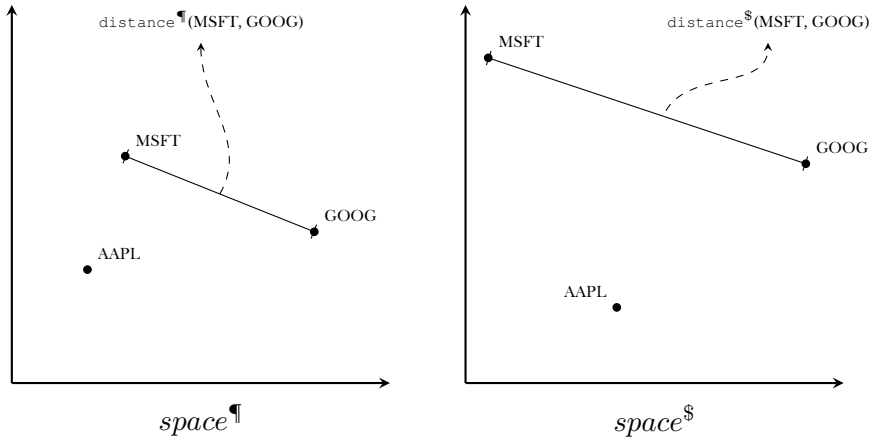


Figure 4.1: Illustration of two implicitly defined spaces: $space^{¶}$ and $space^{\$}$. The solid lines refer to the distance between two stocks, GOOG (stock symbol of Google Inc) and MSFT (stock symbol of Microsoft Inc).

market trading articles. Data Acquisition is already well addressed by the mature web crawler techniques mentioned in Chapter 3. Here we focus on two key components: **Distance Aggregation** and **Relationship Quantification**.

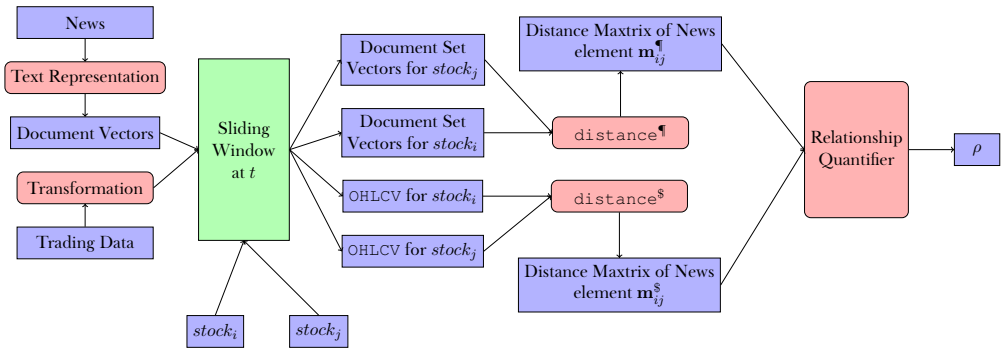


Figure 4.2: An overview of the proposed framework. Blue rectangles refer to input, output or intermediate data. Red rectangles stand for the key components of the framework. The framework produces a value ρ for given stock i and j for sliding window at time t. Depending on the Relationship Quantifier, the ρ may come with a *p-value* to indicate if ρ is statistically significant or not.

Distance Aggregation: The framework is based on measuring the degrees of difference between pairs of stocks. The dissimilarity is measured using two sources of information: financial news and financial time series. In other words, we implicitly define two hyper-spaces by aggregating the distances between pairs of stocks. Later in this chapter, we use the paragraph symbol '¶' as a superscript to label news, and the dollar symbol '$' for time series. Figure 4.1 presents

the idea of establishing the two hyper-spaces. It is worth mentioning that all distance calculations are based on vectors; the points shown in Figure 4.1 are actually vectorised information from news or time series.

Various approaches have been proposed to solve the text similarity problem. For the purpose of illustrating the flexibility of our framework, the recent, popular Word2Vec and the classic unigram BoW -TFIDF algorithms were chosen for the document representations, and a variation of WMD , document mover's distance, was used for distance measurement between two sets of texts. For better capturing of patterns, we chose DTW as the similarity metric between time series.

Relationship Quantification: Once the distances are obtained, the Relationship Quantification component (Relationship Quantifier) needs to perform a series of statistical tests to quantify the relationship between $distance^{¶}$ and $distance^{\$}$. In this research we present the Mantel test (see section 4.4.1) as one relationship quantifier. Since a number of studies doubt the performance of the Mantel test, we also provide results using Spearman's rank-order correlation (section 4.4.2).

4.3 Distance Aggregation: News and Time Series

4.3.1 Stock News Representations

The first step is to represent each stock news article as a single vector. Word2Vec and Bag-of-Words with the TFIDF weighting scheme (BoW -TFIDF) were selected as the baselines for the purpose of comparison of text representations.

Word2Vec Representation

After investigation of related studies of all existing pre-trained Word2Vec models (Komninos and Manandhar, 2016; Le et al., 2017; Reimers and Gurevych, 2016), we chose the one

proposed by Komninos and Manandhar (2016) for its vocabulary size, reported outstanding performance on multiple NLP tasks ((Reimers and Gurevych, 2016)) and the context-based model (*dep* model) proposed by the authors can help to deal with OOV words (explained below).

The first model we used is the *dep* model described in (Komninos and Manandhar, 2016), denoted as W2V*dep* . The embedding of each word in a sentence is the average of its associated dependency-based embeddings. The major reason for choosing this model was dealing with the out-of-vocabulary (OOV) problem. Some OOV words may appear frequently in a short period of time (e.g. breaking events or new technologies). Although the embeddings are trained using the huge Wikipedia corpus which covers a fair amount of words, there are still a lot of invented words, product names or company names that the Wikipedia corpus can not cover over time. Some of them are important words but they do not appear in the embedding vocabulary, for example, `dieselgate` (event name of the Volkswagen cheating scandal), `taptic` (Apple's famous technology used in its phones), `acorda` (a company name). Figure 4.3 illustrates a dependency graph with OOV word `taptic`[1]. When calculating embedding of `taptic`, all its contexts are not OOV therefore an OOV word will always get its embedding from its contexts. Note that if there is more than one OOV word in a sentence, the sentence is ignored in our case.

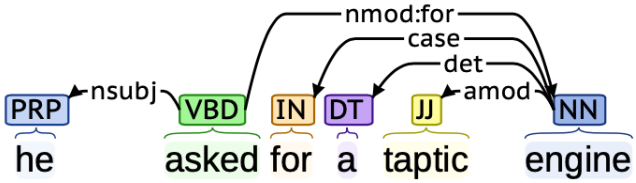


Figure 4.3: An OOV example sentence, where `taptic` is the OOV word. The contexts are `engine`, `det_a`, `case_for`, `nmod for_inv_asked` when querying vectors from the pretrained Word2Vec model.

[1] *taptic engine* is a vibration component in Apple's 7th generation iPhone

Formally, given a document $\mathbf{D}$ with N words $\{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N\}$, we denote embedding of w_i as $\mathbf{v}_{\mathbf{W}_i}$, where $i \in \{1, \cdots, N\}$ and a set of dependency context features acquired after dependency parsing for $\mathbf{w}_i$, denoted as $\{d_1^i, d_2^i, \cdots, d_C^i\}$, and corresponding embeddings $\{\mathbf{v}_{d_1^i}, \mathbf{v}_{d_2^i}, \cdots, \mathbf{v}_{d_C^i}\}$. The dependency parsing is performed using SpaCy 2.0 dependency parser (Honnibal and Montani, 2017), it also uses Universal Dependency Relations (De Marneffe et al., 2014), as in the model proposed in (Komninos and Manandhar, 2016).

$$\mathbf{x}_{\mathbf{w}_i} = \frac{1}{C}\sum_{c=1}^{C} \mathbf{v}_{d_c^i}, \text{where } \mathbf{v}_{d_c^i} = \mathbf{x}_{\mathbf{w}_c} \text{ if } \mathbf{w}_c \in \text{OOV}. \tag{4.1}$$

This is equivalent to calculate embedding of OOV word first and then proceed to other words.

The second model we used simply used only the word embeddings (equation 4.2), which corresponds to the *Words* approach in (Komninos and Manandhar, 2016), denoted as W2V$words$.

$$\mathbf{x}_{\mathbf{w}_i} = \mathbf{v}_{\mathbf{w}_i}, \text{where } \mathbf{w}_i \notin \text{OOV}. \tag{4.2}$$

Note that in this case, we no longer take OOV words into account.

Document Vector According to previous studies (Kenter et al., 2016), averaging the embeddings of words in a sentence is an efficient and "surprisingly successful" method of achieving sentence embedding. We represented each document by taking the average embeddings of all words in it, where

$$\mathbf{v}_{\mathbf{D}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_{\mathbf{w}_n} \tag{4.3}$$

BoW TFIDF Representation

A BoW unigram model with a TFIDF weighting scheme was applied to all collected online news. A news article was then represented using a vector with each column representing a token in it and the value equal to the TFIDF weight. A set of news articles was therefore

represented as a matrix in which each row was a vector of a news article. We also used the average of all vectors of words in a document as the document vector.

To clarify, we pre-processed all text documents in the following way: first, the text was tokenised, i.e. split into separate words or punctuation symbols. Then we removed all punctuation and stop words, essentially all pronouns, prepositions, conjunctions and a few very common verbs. The remaining words were lemmatised, i.e. replaced by their standard entry in the dictionary. All urls were then mapped to the same string `URL`, email addresses were mapped to the string `EMAIL` and numbers were mapped to `NUM`.

4.3.2 Similarity between Sets of News: Document Mover's Distance

To restate our problem, each stock may have more than one piece of news during a given period. The distance measure needs to take account of two sets with different numbers of vectors. For example, company A has 65 news items published, while company B only has 12 news items during the same week. Therefore, the distance needs to be calculated between two vectors $65 \times K$ and $12 \times K$, where K is the number of dimensions for each news item.

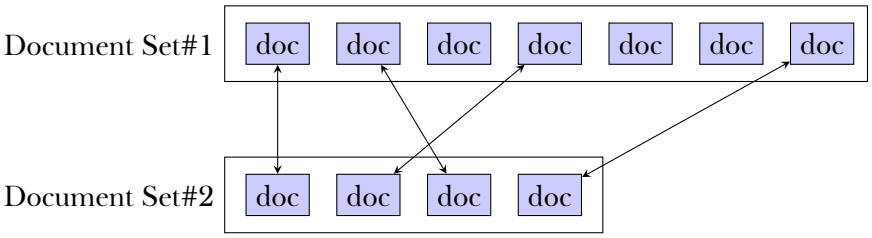


Figure 4.4: Illustration of the document mover's distance between two document sets.

Our approach leverages the Word Mover's Distance (WMD), a recent document-level similarity metric proposed by Kusner et al. (2015). As introduced in Section 2.4.2, the WMD treats dissimilarity measuring as a transportation problem. The travel cost between two words is the Euclidean distance between two vectors obtained by a Word2Vec model. Similarly, our approach first represents documents as a single vector by averaging over each dimension. Therefore, a document in our approach is equivalent to a word in the WMD , and a set of documents

can be treated as a sentence in the WMD metric. Since a document does not occur more than once in a document set, the weighting coefficient for each document is the same — simply the inverse document frequency with respect to the size of the document set.

4.3.3 Stock Price: Transformation and Distance Measuring

The stock price of a company consists of multiple time series, i.e. open, high, low, close and volume of trades (OHLCV, for abbreviations, see 2.1.2). However, this does not provide enough information or a direct description which can be used for further analysis of the financial performance of a stock. We opted to aggregate the price data to generate three extra time series following the equations below:

$$r_t = \frac{o_t - p_t}{o_t} \tag{4.4}$$

$$r'_t = \frac{a_t - a_{t-1}}{a_{t-1}} \tag{4.5}$$

$$c_t = \frac{h_t - l_t}{o_t} \tag{4.6}$$

where o, h, l and p are time series representations of open, high, low and close, a refers to the *adjusted close price*[2] (ADJ for short). r is the *intraday return* (IRT), which measures the intraday profit ratio gained by a stock. r' stands for the *overnight return* (ORT), which measures the overnight profit ratio of a stock and c is *change* (CHG) in a day, which is a metric of volatility.

Our research focuses on five time series: ADJ, VOL, CHG, IRT and ORT. These financial time series reveal a stock's performance from different perspectives. For example, ADJ gives the trend of the stock price, VOL can provide information on the liquidity and stability of a position (support or resistant level as usually seen in technical analysis). Intraday return and

[2]An adjusted closing price is a stock's closing price amended to the price taking dividends, splits, right offers etc. into account so that the price is comparable with previous closing prices.

overnight return give direct information on how a stock performs during the day or overnight. DTW , described in the previous chapter, is used to calculate the distance for a pair of time series.

4.4 Relationship Quantifying: Mantel Test and Spearman's Rank-Order Correlation

Online news data, which is a type of natural language text, is mostly published without a fixed time interval. The themes of a piece of news cover a lot of topics, new products, financial performance, analysts' views, or extreme events, such as recall, acquisition or a lawsuit. Although the original data is generated 'tick-by-tick', financial time series are usually regenerated and aligned to a specific time interval according to demand, with ranges from 1 minute to monthly. Moreover, unlike time series data, online news information is in the form of natural text, which is highly unstructured and the effective extraction of this information is either too expensive or unfeasible.

4.4.1 Mantel Test

The Mantel test is a statistical test to determine the correlation between two pairwise distance matrices with the same rank. The advantage, similar to tree kernel (Moschitti, 2006), is that implicitly defining the distance between a pair of instances can be easier than explicitly defining a space.

Mantel (1967) introduced the Mantel test based on permutations. The test is able to assess the significance of the association between two matrices of distances relative to the same pairs of individuals or demes.[3]

Formally, given a set of l objects (e.g. animals belonging to the same species) and two

[3]To clarify, we use the simple Mantel test in this research, not the partial Mantel test proposed in (Sokal, 1979).

different aspects of observation, e.g. weight and location, dissimilarities between each pair of objects can be measured, and two matrices, $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be generated accordingly. The Mantel test assumes that if there is a relationship between two distance matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, then the sum of the cross product between the two matrices will be relatively high. Shuffling rows and columns of one matrix will affect such relationships, so that the sum will consequently change to a smaller value. The Mantel approach can also be used to test a hypothesis or a model. In the model testing approach, one matrix contains response data, while the other contains a representation of an a priori model to test. The model matrix thus represents the alternative hypothesis for the test. If significant Mantel statistics are found, they provide some support for the model (Buttigieg and Ramette, 2014).

Mantel statistics are tested for significance of the Mantel correlation by repeatedly switching indices of rows and columns of a random pair of objects, called *permutation* (see Figure 4.5). The Mantel correlation, defined in Equation 4.7, is calculated after each permutation.

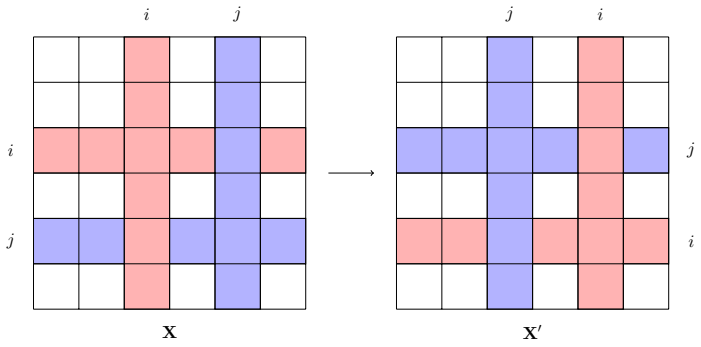


Figure 4.5: Illustration of *permutation* operation on matrix $\boldsymbol{X}$. The permutation randomly switches object i and j so that corresponding elements in the distance matrix switch positions accordingly.

$$\rho_M = \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{x_{ij} - \bar{x}}{\delta_x} \frac{y_{ij} - \bar{y}}{\delta_y} \tag{4.7}$$

where x_{ij} and y_{ij} are the distances (dissimilarity) between objects i and j, measured with respect to two different approaches to observation (e.g, spatial and temporal). $\boldsymbol{X}$ and $\boldsymbol{Y}$ have the

dimension of $l \times l$, total $m = l(l-1)/2$ non-redundant distance elements (the elements of the upper or lower triangular exclude the diagonal). Note that the columns and rows of $\boldsymbol{Y}$ are randomly shuffled symmetrically, the observed Mantel correlation after permutation, noted as $\rho_{permuted}$, is compared with the original ρ calculated without permutation. To put it in the form of hypothesis testing, the null hypothesis is:

> $\boldsymbol{H}_0$: The pair-wise distance elements between l objects in matrix $\boldsymbol{X}$ and $\boldsymbol{Y}$, are *not* linearly correlated with each other.

and the alternative hypothesis is

> $\boldsymbol{H}_a$: The pair-wise distance elements between l objects in $\boldsymbol{X}$ and $\boldsymbol{Y}$ are linearly correlated with each other.

The alternative hypothesis $\boldsymbol{H}_a$, in other words, is to confirm that one factor is linearly correlated with another factor. The null hypothesis $\boldsymbol{H}_0$ will be rejected if:

$$p\text{-}value = \frac{Count(|\rho_{permuted}| > |\rho|)}{N} < \alpha \tag{4.8}$$

where α is the degree of confidence, N is the number of permutations that should be made for the hypothesis test, equivalent to the number of samples or observations, which depends on α, typically $1,000$ for $\alpha = 0.05$, 5000 for $\alpha = 0.01$ and $10,000$ if higher precision is required. A two-tailed $p\text{-}value$ is used in this research since the absolute value of the Mantel correlation is tested.

4.4.2 The Spearman's Rank Correlation Coefficient

The Spearman's rank correlation coefficient is a nonparametric test that measures the strength and direction of association between two ranked variables. Given n pairs of observations, the

Algorithm 2 The Mantel Test

Require: $\boldsymbol{X}$, $\boldsymbol{Y}$, n_{perm}, α
Ensure: $\boldsymbol{X}$, $\boldsymbol{Y}$ are symmetrical and all elements of their diagonals are zeros.
Null hypothesis $\boldsymbol{H}_0$: $\boldsymbol{X}$ and $\boldsymbol{Y}$ are not linearly correlated.
procedure MANTELTEST($\boldsymbol{X}$, $\boldsymbol{Y}$, n_{perm}, α)
    $\rho_0 = \texttt{MantelCorrelation}(\boldsymbol{X}, \boldsymbol{Y})$
    $i = 0$
    $samples = \langle \rho_0 \rangle$
    while $i \leqslant n_{perm}$ **do**
        $\boldsymbol{Y}_{perm} \leftarrow \texttt{Permutate}(\boldsymbol{Y})$
        $\rho_{perm} \leftarrow \texttt{MantelCorrelation}(\boldsymbol{X}, \boldsymbol{Y}_{perm})$
        $i \leftarrow i + 1$
        $samples[i + 1] \leftarrow \rho_{perm}$
    end while
    $p\text{-}value = count(|\rho_{perm}| > |\rho_0|)/n_{perm}$
    if $p\text{-}value < \alpha$ **then**
        Reject Null hypothesis $\boldsymbol{H}_0$
    else
        Accept Null hypothesis $\boldsymbol{H}_0$
    end if
end procedure

function PERMUTATE($\boldsymbol{M}$)
    $j \leftarrow$ random column index
    $k \leftarrow$ random column index, and $k \neq j$
    $SwitchColumns(\boldsymbol{M}, j, k)$
    $SwitchRows(\boldsymbol{M}, j, k)$
    return $\boldsymbol{M}$
end function

function MANTELCORRELATION($\boldsymbol{X}$, $\boldsymbol{Y}$)
    $$\rho \leftarrow \sum_{j=1}^{m} \sum_{k=1}^{m} \frac{x_{jk} - \bar{x}}{\delta_x} \frac{y_{jk} - \bar{y}}{\delta_y}$$
    return ρ
end function

Spearman's rank correlation coefficient r_S can be calculated after sorting the observations separately in ascending order.

$$r_S = \frac{cov(rg_u, rg_v)}{\sigma_{rg_u}\sigma_{rg_v}} \tag{4.9}$$

where rg_u and rg_v are the converted rankings of variable u and v. $cov(rg_u, rg_v)$ is the covariance. σ_{rg_u} and σ_{rg_v} are the standard deviations of the ranking of variables u and v.

Null hypothesis:

$\boldsymbol{H}_0$: $\rho_S = 0$ so that there is no population correlation between ranks.

Alternative hypothesis:

$\boldsymbol{H}_a$: $\rho_S > 0$ or $\rho_S < 0$ so that there is correlation between ranks.

When the $\boldsymbol{H}_0$ is valid, r_S will be approximately a normal distribution, where

$$\mu_{r_S} = 0$$
$$\sigma_{r_S} = \sqrt{\frac{1}{n-1}}$$

The *p-value* can be obtained from the random variable Z which is approximately normal distributed.

$$Z = \frac{r_S - \mu_{r_S}}{\sigma_{r_S}} = r_S\sqrt{n-1} \tag{4.10}$$

4.4.3 Sliding Window Approach

We used the sliding window approach to capture the dynamic relationship between stock news and trading data. Figure 4.6 shows the arrangement of a sliding window of news and a sliding window of time series.

As financial markets do not trade every day, the sliding windows were generated according to trading days instead of calendar days. To better capture information from the news, we used

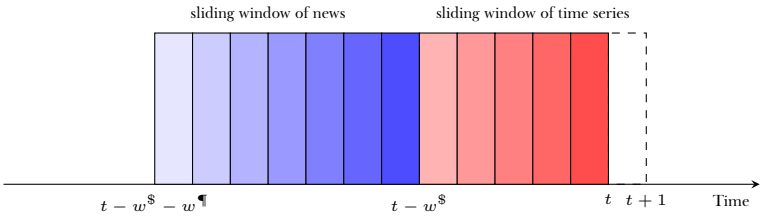


Figure 4.6: A pair of sliding windows at time t: The time axis represents trading days over time. The sliding window of news in blue has $w^{¶}$ days and the sliding window of the time series includes $w^{\$}$ trading days. The two sliding windows do not overlap. Every pair of sliding windows goes one trading day ahead of the previous pair.

different sizes of sliding window for news and time series. The time series used a 5-trade-day width in order to cover a complete trading week. The news sliding window has a 7-calendar-day width to cover the whole week before the time series. Since news is always available, no matter if there is trading or not, our task is to see if changes in the $space^{¶}$ are related to changes in the $space^{\$}$. The time series sliding window follows the news sliding window.

4.5 Description of Data and Experiment

4.5.1 The Data

We collected online news from Yahoo from 1st October 2014 to 30th April 2015. Each news article carries an EST time stamp and the symbol of one or more stocks to which it is related. The 25 stocks were selected as having no more than 5 calendar days with no news about them in the studied period. Very short articles with fewer than 10 words or 100 characters were ignored, leaving a total of 189,151 news items. Stock trading data (OHLCV) from the same period was collected from Yahoo Finance. Table 4.1 and Table 4.2 give an overview of the data used in this research.

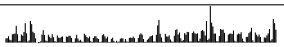
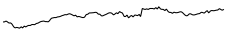
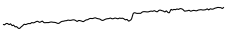
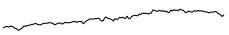
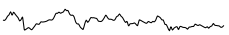
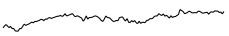
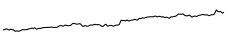
4.5.2 Experimental Design

Previous work (Qu and Kazakov, 2016) has explored the relationship between news and trading time series using a universal sliding window. Here the experiment follows the basic frame-

Table 4.1: Overview of data used in this research

Item	Value
Date	2014-10-1 to 2015-4-30
Number of stocks	25
Number of news items	189,151
Number of calendar days	212
Number of trading days	142

Table 4.2: Information for each company: symbol, name, stock exchange, close price (ADJ) and number of news items per day.

Symbol	Name	ADJ	Number of News per Day
AAPL	Apple Inc.		
AMZN	Amazon.com Inc.		
BA	Boeing Co.		
CMCSA	Comcast Co.		
CSCO	Cisco Systems, Inc.		
CVX	Chevron Co.		
DIS	Walt Disney Co.		
EBAY	eBay Inc.		
FB	Facebook Common Stock		
GOOG	Alphabet Inc. Class C		
GOOGL	Alphabet Inc. Class A		
GS	Goldman Sachs Group Inc.		
HD	Home Depot Inc.		
INTC	Intel Co.		
JPM	JPMorgan Chase & Co.		
KO	The Coca-Cola Co.		
MSFT	Microsoft Co.		
NFLX	Netflix, Inc.		
NKE	Nike Inc.		
SBUX	Starbucks Co.		
T	AT&T Inc.		
TSLA	Tesla Motors Inc.		
VZ	Verizon Comm. Inc.		
WMT	Wal-Mart Stores, Inc.		
YHOO	Yahoo! Inc.		

work used in the published study, but with several modifications. Firstly, the aforementioned dual-sliding-window approach put the news ahead of the time series, this helps in discovering the potential 'causal' relationship in later sections. Secondly, TFIDF and Word2Vec representations are applied to the textual data properly, in particular, the Word2Vec representation used here keeps the information at the document level as much as possible. Thirdly, we use different distance metrics, WMD and DTW , for both the textual data and the time series. The biggest advantage is that WMD and DTW break distance between the two sets down to the sample level so that the distance is more accurate.

To compare the ability to capture the financial volatility dynamic, both TFIDF and Word2Vec are used. The aforementioned five time series are all included for the purpose of comparison. The experiment proceeds as follows:

1. Select a textual representation (W2Vdep , W2V$words$ or TFIDF in this research) and one time series, and in this research we use one of adjusted close(ADJ), change (CHG), volume (VOL), intraday return (IRT) or overnight return (ORT) (see subsection 4.3.3).

2. Pre-process news articles and time series as described in Sections 4.3.2 and 4.3.3.

3. For each pair of sliding windows (see Figure 4.6), calculate pairwise distances between all combinations of stocks. Thus, we have two distance matrices $\boldsymbol{M}^{¶}$ and $\boldsymbol{M}^{\$}$ (or distance sequences for Spearman's Rank-Order Correlation).

4. For each pair of sliding windows, perform the Mantel test (or Spearman's Rank-Order Correlation) and output ρ_M (or ρ_S for Spearman's Rank-Order Correlation) and $p\text{-}value$.

5. Repeat from Step 1 with a different selection of textual representation and time series.

4.5.3 Evaluation: Portfolio Selection

The Mantel correlation reflects the linear correlation between the stocks in the *space*¶ and the *space*$. Through the calculation of the Mantel correlation (Algorithm 2), it can be inferred that certain stocks affect the Mantel correlation more than others. These stocks warrant further attention and to find them we exploit an approach called *leave-one-out* (LOO), which follows the idea of *weighing smoke*, explained below.

Weighing Smoke

Assume that we are given a cigarette and need to weigh the mass of the smoke it produces. Physically measuring the weight of the smoke would be a very challenging task. Instead of directly weighing the smoke, we can subtract the cigarette with the ash left.

LOO treats individual stock as the smoke. LOO removes the i-th stock from the stock pool and performs the relationship quantification between $\boldsymbol{M}^{¶}_{\text{-}i}$ and $\boldsymbol{M}^{\$}_{\text{-}i}$. The subscript '-$i$' means excluding the i-th stock. The 'weight of smoke' or change in the correlation δ_i can be obtained by subtracting $\rho_{\text{-}i}$ from ρ.

$$\delta_i = \rho - \rho_{\text{-}i} \tag{4.11}$$

where

$$\rho = \texttt{RelationshipQuantify}(\boldsymbol{M}^{¶}, \boldsymbol{M}^{\$}),$$

$$\rho_{\text{-}i} = \texttt{RelationshipQuantify}(\boldsymbol{M}^{¶}_{\text{-}i}, \boldsymbol{M}^{\$}_{\text{-}i})$$

`RelationshipQuantify` can be either the Mantel test or Spearman's Rank-Order Correlation.

Finding the Winner using δ

δ is used as a technical indicator that measures the correlation contribution of target stock to the overall portfolio. How is this indicator related to stock performance? In this research, intraday return on day $t+1$ is used as the performance metric for stocks. We performed a series of Pearson's Correlation tests to see if this indicator can help in selecting stocks. To clarify, here the Pearson's Correlation is not a relationship quantifier but only detects if the intraday return on $t+1$ day has a monic relation with δ. Another proof of effectiveness is shown in Table 4.4.

Trading Strategies

For the purpose of illustrating the effectiveness of our framework, we use an extremely simple trading strategy using the indicator described in the previous section to assist portfolio selection. After each trading day the strategy selects the stock with the largest δ, buys the stock at next market opening and sells the stock before market closing. The daily return of the portfolio r_{t+1} can be approximated using Equation 4.12.

$$r_{t+1} = r_{t+1}^{(n)} \tag{4.12}$$

where

$$n = arg\max_i \{\delta_t^{(i)}\}_{i\in\{1,\cdots,N\}}$$

n is the index of the selected stock and N is the number of stocks in the portfolio.

The two benchmarks are **Best Stock** (BS) and **Uniform Constant Rebalance** (UCR). BS is a strong baseline strategy which leverages market momentum. It can be briefly described as putting all capital into the stock with the best financial performance. This is a strategy solely based on time series. In this research we chose a 5-day average intraday return as the metric to measure the performance of stocks. Therefore, if stock A has the highest 5-day-average

intraday return on day t, strategy BS will sell any valid position and buy stock A on day $t+1$ using all capital.

$$r_{t+1} = r_{t+1}^{(n)} \tag{4.13}$$

where

$$n = arg\max_i \{\bar{r}_t^{(i)}\}_{i\in\{1,\cdots,N\}}$$

$\bar{r}$ is the 5-day-average intraday return.

The word 'uniform' in UCR refers to an equal share of capital for every stock. UCR adjusts stocks and balances the capital in stocks so that they share the same portion of the total capital after every trading day (Li and Hoi, 2012). The daily return of a UCR portfolio can be calculated by using the average return of all stocks in the portfolio.

$$r_{t+1} = \frac{\sum_{j=i}^{N} r_t^{(j)}}{N} \tag{4.14}$$

Evaluation Metrics

To determine the success of a trading strategy, the accumulative portfolio value (APV) is used for a straightforward comparison of ability to produce profit.

$$APV_t = \frac{p_t}{p_0} \tag{4.15}$$

where p_t is the value of assets plus funds in the portfolio at time t, and p_0 is the initial fund allocated to the strategy. The higher the APV, the more profit a strategy produces.

However, APV does not give credit to those strategies which drop less when risks happen, but with a slightly lower profit. Therefore, the Sharpe Ratio (SR) is proposed to measure the ratio between the risk and the potential profit a strategy can make.

$$SR = \frac{\mathbb{E}[r - r_F]}{\sqrt{var[r - r_F]}} \tag{4.16}$$

where r is the periodic return and r_F is the return of the risk-free asset. We assume the risk-free asset is cash, therefore $r_F = 0$ in this research.

SR takes the volatility of the portfolio into account, but treats the upwards and downwards trends equally, even though upwards movements increase the APV of a portfolio. Maximum Drawdown (MD) is proposed to give a specific measure of how much loss at time t a portfolio can make since its maximum APV t.

$$MD_t = \frac{max_0^\tau(\{APV_i\}) - min_\tau^t(\{APV_i\})}{max_0^\tau(\{APV_i\})} \tag{4.17}$$

where max_0^τ means the maximum between 0 and tau, and min_τ^t means the minimum between τ and t.

Annualised Return (AR) is a commonly used metric to estimate the rate of return for an investment each year.

$$AR = (\frac{p_T - p_0}{p_0})^{\frac{252}{T}} \tag{4.18}$$

where T refers to the number of trading days from the beginning of a strategy. 252 is a universal number of trading days in a whole calendar year.

4.6 Results and Discussions

4.6.1 Results of Quantified Relationship

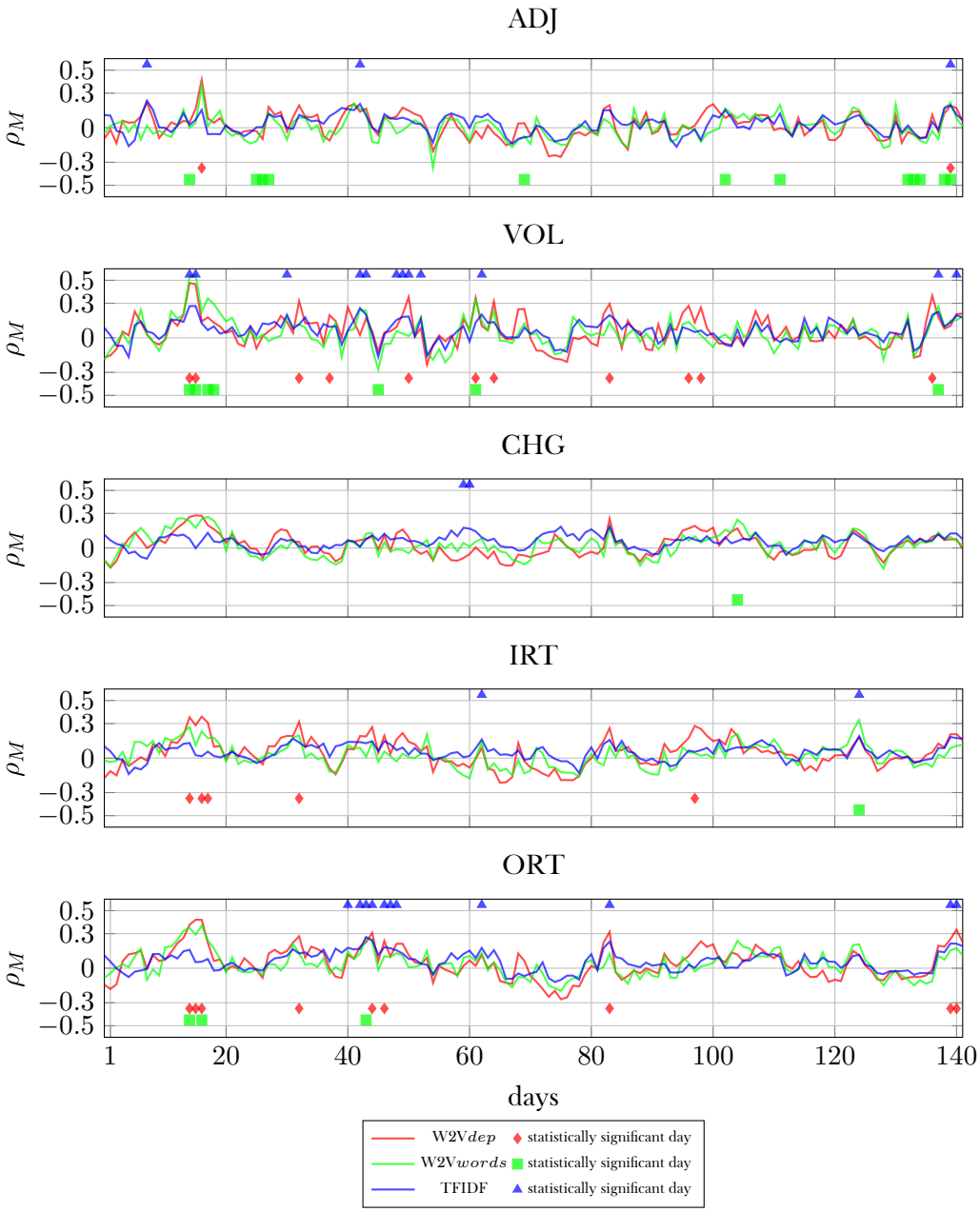


Figure 4.7: Mantel correlations ρ_M of time series. The red solid line and diamonds refer to tests on distance based on W2Vdep , the green solid line and squares refer to those based on W2V$words$ and the blue solid line and triangles refer to those based on TFIDF .

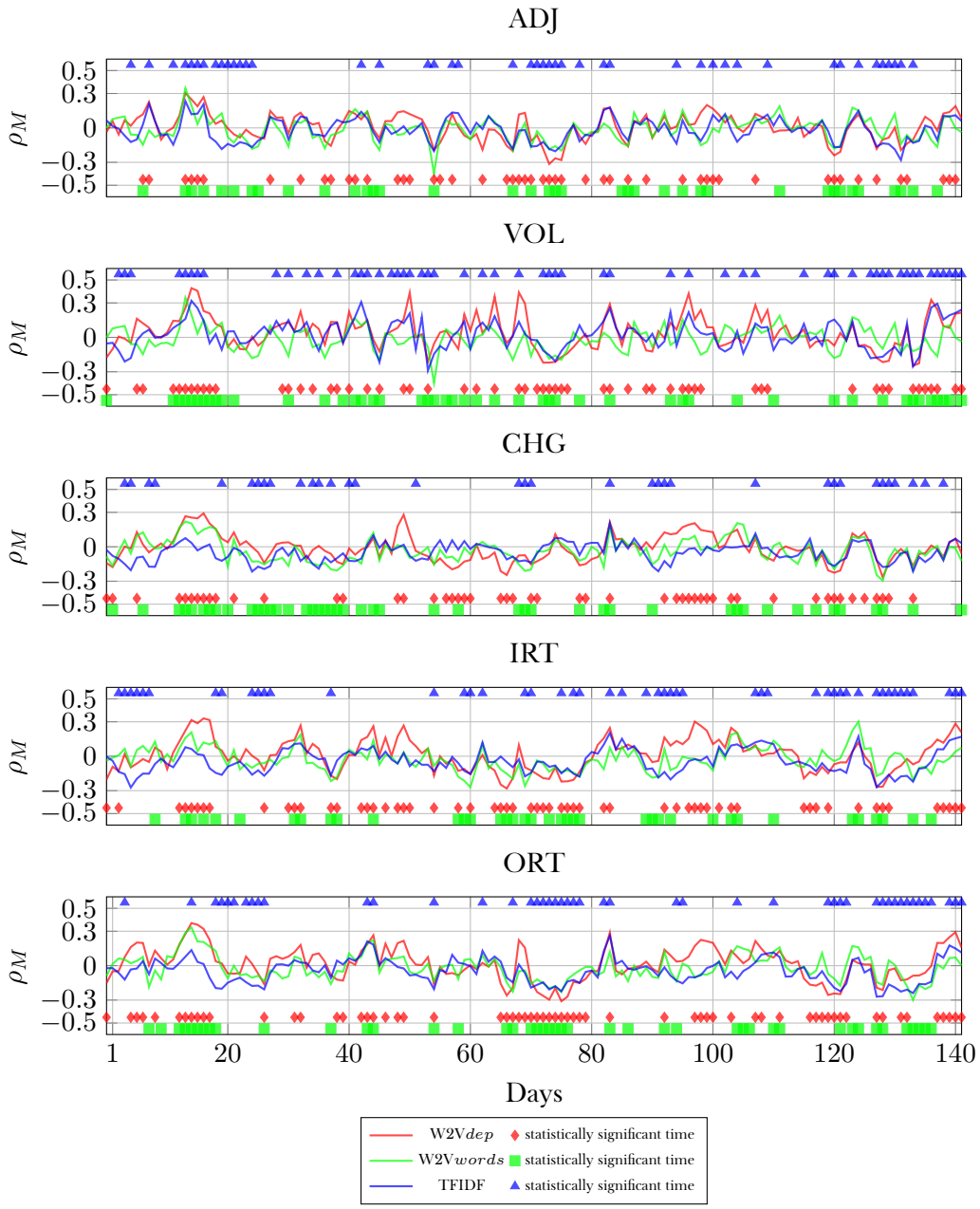


Figure 4.8: Spearman correlations ρ_S of time series. The red solid line and diamonds refer to tests on distance based on W2V*dep* , the green solid line and squares refer to those based on W2V*words* and the blue solid line and triangles refer to those based on TFIDF .

Figure 4.7 suggests low to moderate levels of Mantel correlation between news and financial performance, which for the VOL (volume) and W2V*dep* combination reaches values around 0.47, while W2V*words* even reaches 0.55. Figure 4.8 also presents low to moderate levels of correlation coefficient between news and financial time series, which for the ADJ (adjusted close) and W2V*dep* combination reaches values around 0.43.

Importantly, a statistically significant ($p\text{-}value \leqslant 0.05$) high Mantel correlation means the stock pool performance is correlated with the news in the historical 7-day observation window. We do not attempt to expand this topic here, as long as we observe there is a valid correlation between the news and the time series.

Generally speaking, Word2Vec (W2Vdep and W2V$words$) and TFIDF show rather different results. The TFIDF representation tends to give a less volatile correlation than Word2Vec . On the other hand, the general trends of Word2Vec and TFIDF stay the same most of the time. There is no direct evidence from the results to support one being better than the other. But the text representations are further compared in a later evaluation section.

From Figure 4.7 and Figure 4.8 we can see that the general trend of the correlation stays the same in all time series. VOL (trading volume) shows the most volatile curve in both figures.

4.6.2 Results of Leave-One-Out

Table 4.3: The Pearson's Correlation results between next day's intraday return r and δ defined in section 4.5.3 across all stocks. Each number is the portion of trading days when δ and r have a significant linear correlation with a confidence level of 95%.

		Time Series				
Quantifier	**Representation**	ADJ[%]	CHG[%]	VOL[%]	IRT[%]	ORT[%]
Mantel	W2Vdep	10.563	17.606	8.451	17.606	15.493
	W2V$words$	8.450	14.789	5.634	16.901	16.197
	TFIDF	11.972	21.127	7.042	17.606	16.901
Spearman	W2Vdep	10.563	9.155	4.930	12.676	9.155
	W2V$words$	5.634	10.563	7.746	10.563	11.268
	TFIDF	3.521	11.972	4.225	11.972	7.042

Table 4.4: Ratios when selected stock are in next day's top five winners. Each number is the portion of trading days when the selected stock appears in the top five stocks with highest intraday return on the next trading day.

		Time Series				
Quantifier	**Representation**	ADJ[%]	CHG[%]	VOL[%]	IRT[%]	ORT[%]
Mantel	W2Vdep	23.239	28.873	30.986	26.056	22.535
	W2V$words$	20.423	28.873	23.239	23.944	23.239
	TFIDF	21.127	26.761	31.690	22.535	18.310
Spearman	W2Vdep	26.056	28.873	30.986	30.986	29.577
	W2V$words$	21.831	27.465	22.535	25.352	24.648
	TFIDF	18.310	31.690	28.169	28.873	25.352

Table 4.3 illustrates that δ provides relatively significant indicative power for selecting winner stocks with higher intraday return from the portfolio on the next trading day. Almost all combinations beat the random choice hit ratio of 20% (5 out of 25) except Spearman-TFIDF and Mantel-TFIDF . Interestingly, δ produced using VOL (volume) gives the highest hit ratio among all time series using Mantel as the quantifier. This potentially confirms the previous result of the observed quantified relations in which VOL had the highest volatility.

4.6.3 Results of Portfolio Selection

Table 4.5: Financial performances of trading strategies. There are three groups of participants. The first group shows results using Mantel's Correlation as the Relationship Quantifier; the second group shows results using Spearman's Rank-Order Correlation as the Relationship Quantifier; in the last group BS and UCR are baselines, where BS is the Best Stock strategy, and UCR is the Uniform-Constant Rebalance strategy. fAPV refers to final-Accumulated Portfolio Value; SR stands for Sharpe Ratio; MD is Maximum Drawdown; AR is the Annualised Return. Bold numbers labeled with '*' are the best among the participants.

News	Time series	fAPV	SR	MD [%]	AR [%]
		Mantel			
W2V*dep*	ADJ	1.395	2.644	9.980	80.437
	CHG	***1.478**	2.336	15.017	***100.155**
	VOL	1.339	2.299	10.641	67.951
	IRT	1.134	0.897	11.837	24.938
	ORT	1.317	1.897	12.763	62.944
W2V*words*	ADJ	1.052	0.579	10.658	9.466
	CHG	1.165	1.122	13.448	31.053
	VOL	1.308	2.326	8.559	61.016
	IRT	1.236	1.492	10.927	45.596
	ORT	1.264	1.956	11.835	51.520
TFIDF	ADJ	0.960	-0.273	10.911	-7.011
	CHG	1.068	0.592	17.456	12.464
	VOL	1.061	0.595	16.260	11.017
	IRT	0.929	-0.423	14.402	-12.242
	ORT	0.972	-0.121	14.805	-4.972
		Spearman			
W2V*dep*	ADJ	1.164	1.471	8.233	30.870
	CHG	1.027	0.308	16.311	4.850
	VOL	1.413	***2.992**	***6.661**	84.730
	IRT	1.132	0.940	19.347	24.642
	ORT	1.209	1.483	11.461	40.146
W2V*words*	ADJ	1.131	1.175	9.618	24.484
	CHG	1.241	1.605	14.673	46.683
	VOL	1.143	1.141	11.186	26.865
	IRT	1.339	2.112	8.109	67.875
	ORT	1.241	1.674	10.440	46.648
TFIDF	ADJ	1.151	1.368	13.937	28.268
	CHG	1.345	1.965	10.271	69.167
	VOL	1.261	1.879	15.370	50.873
	IRT	1.266	1.666	9.232	51.922
	ORT	1.339	2.244	9.857	67.930
		Baseline			
BS		1.271	2.148	12.366	53.012
UCR		1.097	1.530	7.007	17.868

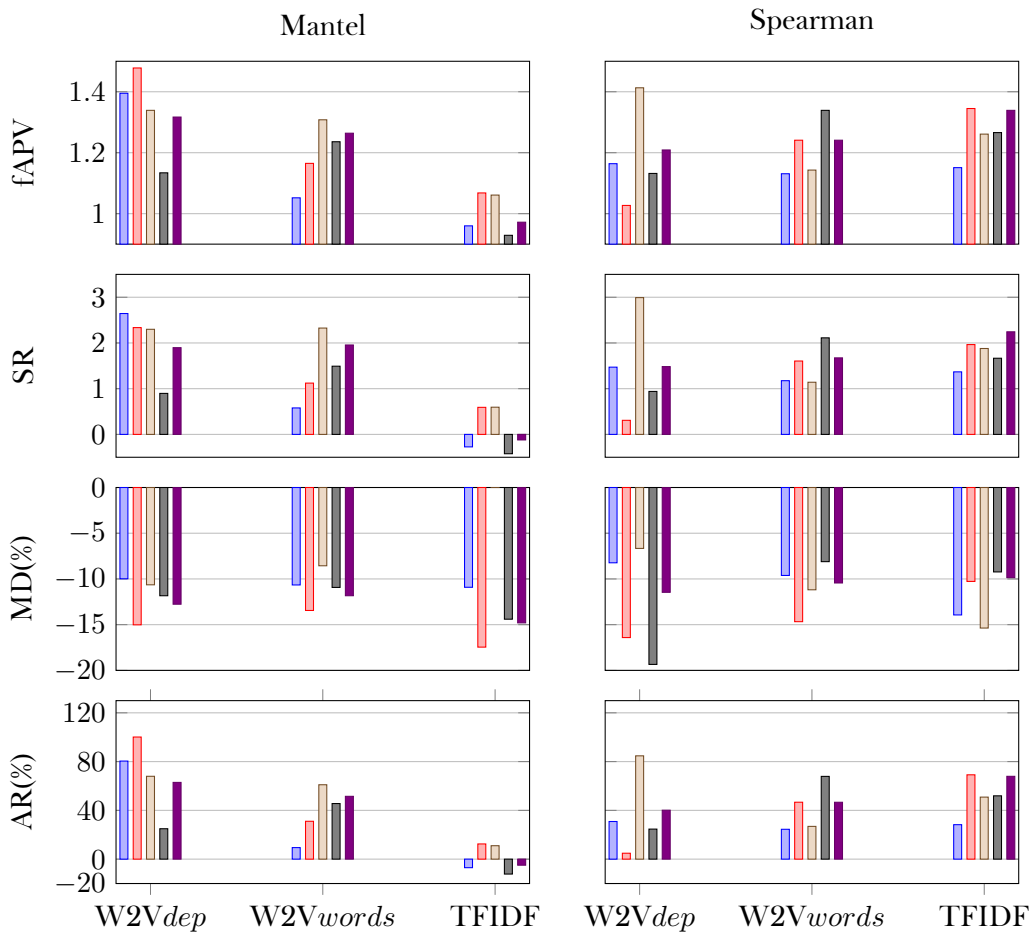


Figure 4.9: Comparsion of financial performances of trading strategies.

Table 4.5 and Figure 4.9 illustrate the overall comparison of all trading strategies. Spearman-Word2Vec -VOL gives the best SR (Sharpe ratio) at 2.992, however Mantel-Word2Vec combinations generally outperform the rest, especially using ADJ, CHG and VOL; their SR varies between 2.299 and 2.644. Remarkably, the Word2Vec -VOL combinations with both the Mantel and Spearman quantifier yields a high SR above 2.2.

Figure 4.10 and 4.11 show the APV curve over time for all strategies. The Mantel-Word2Vec combinations give better curves than the rest and most curves outperform the BS strategy most times. Mantel-TFIDF shows a rather lower performance than the baselines over time compared with the Word2Vec representation. On the other hand, Spearman combinations show diverse curves but most strategies vary between BS and UCR, except the curve of

VOL.

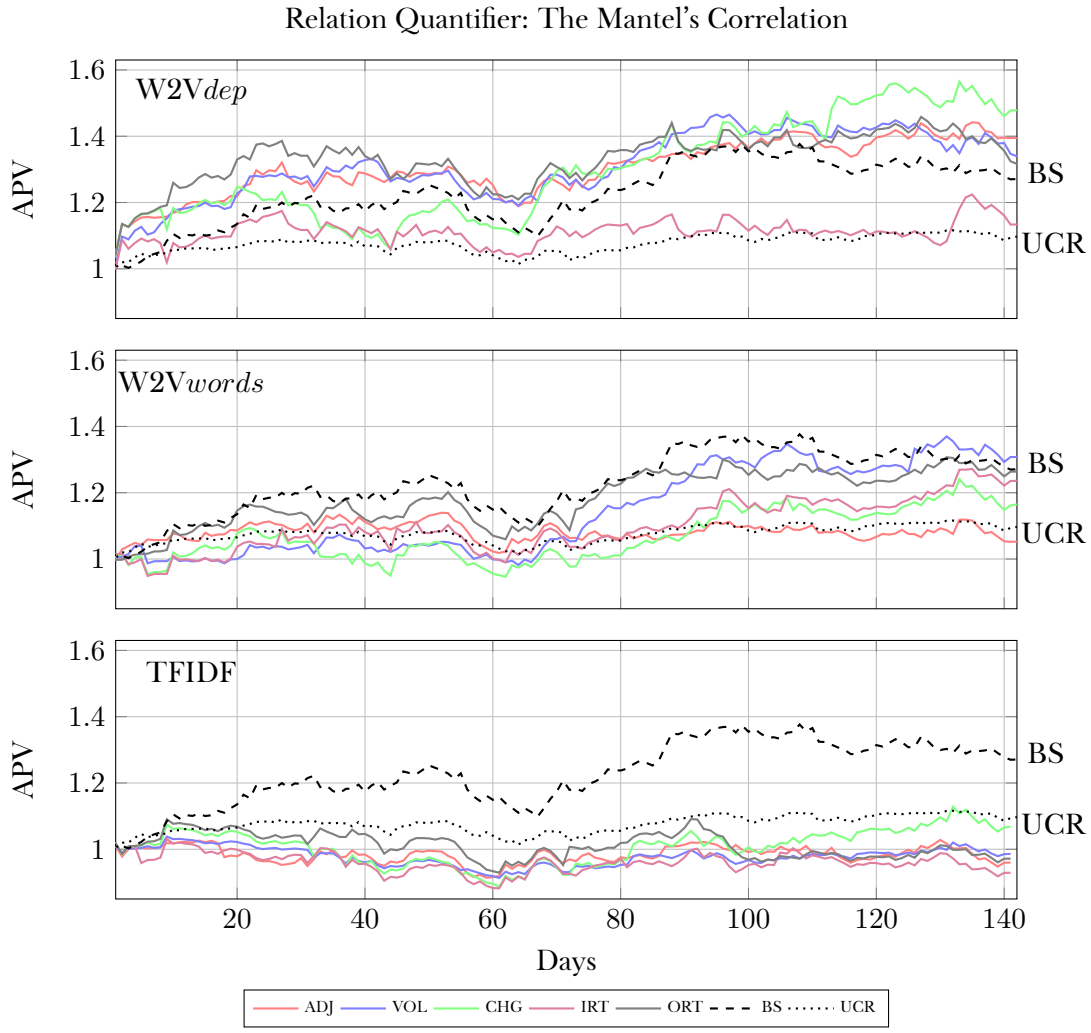


Figure 4.10: Accumulative Portfolio Values (APV) of strategies using the Mantel test as Relationship Quantifier. The top figure shows those strategies using Word2Vec for the Mantel test and the bottom figure shows those using TFIDF . Both figures contain the baseline strategies, BS (dashed line) and UCR (dotted line).

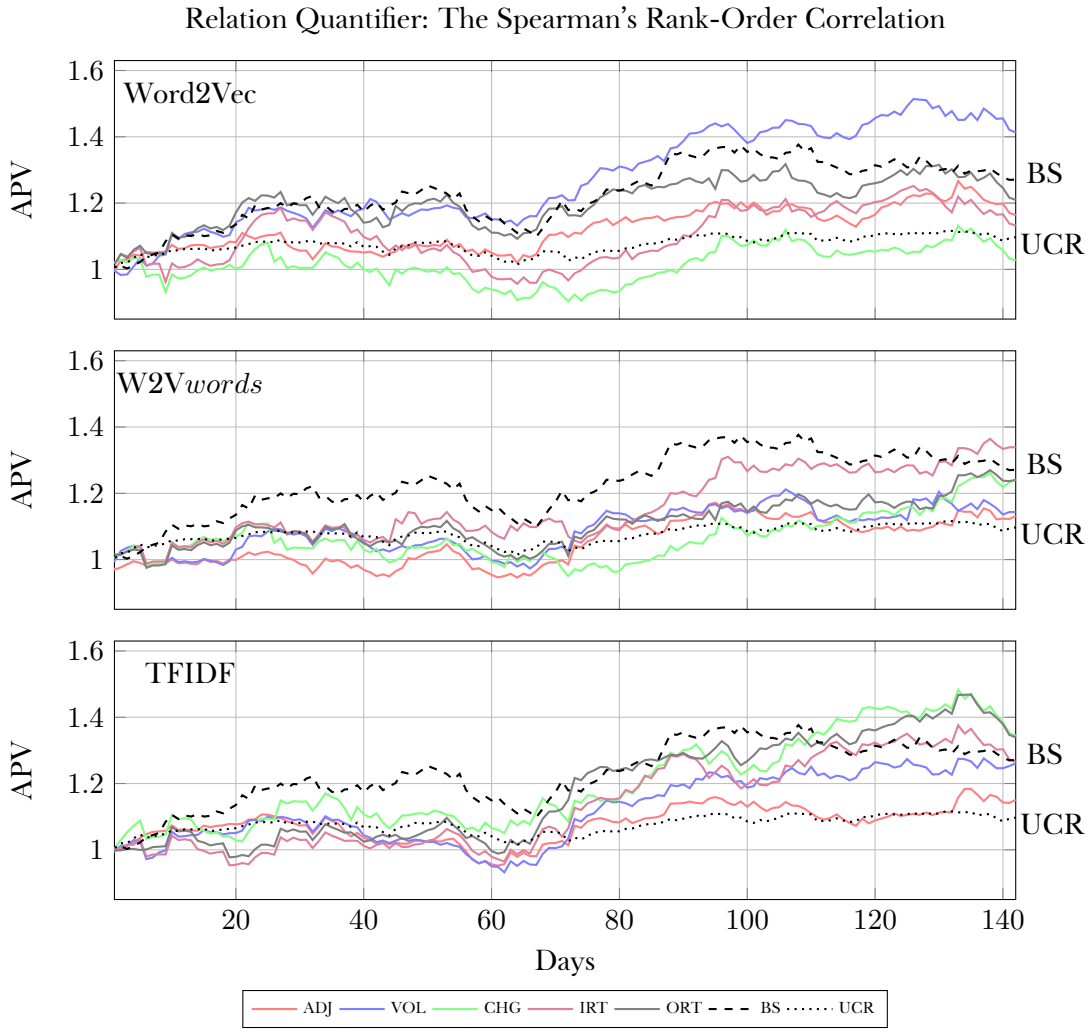


Figure 4.11: Accumulative Portfolio Values (APV) of strategies using Spearman's Rank-Order Correlation Coefficient asthe Relationship Quantifier. The top figure shows those strategies using Word2Vec and the bottom figure shows those using TFIDF . Both figures contain the baseline strategies, BS (dashed line) and UCR (dotted line).

4.6.3.1 Results and Discussion

The main goal of the evaluation experiment was to illustrate the effectiveness of the quantified relationship using the proposed framework. We compared the performance of combinations of text representations (W2V*dep* , W2V*words* and TFIDF), time series (ADJ, CHG, VOL, IRT, ORT) and also relationship quantifiers (Mantel's Correlation and Spearman's Rank-Order Correlation).

There were significant differences between Word2Vec and TFIDF with regard to quantifier

selection between the Mantel correlation and Spearman's Rank-Order correlation. Strategies using Word2Vec outperformed those with TFIDF , which is in agreement with previous research by Kusner et al. (2015) that Word2Vec with WMD distance produces better clusters than TFIDF .

When using the Mantel correlation W2V*dep* and W2V*words* perform similarly when using trading volume (VOL) and overnight return (ORT), and for the other scenarios W2V*dep* outperforms W2V*words* . This indicates that context information is helpful for our task.

The results also indicate that the selection of the Relation Quantifier leads to different strategies being used. The Mantel test relies on distance measurements between the target objects, therefore, text representation with better performance on the clustering tasks leads to precise and reliable quantified relationships using the Mantel test. Spearman's Rank Order Correlation on the other hand focuses on the rankings instead of the precise value of pair-wise distances, therefore, the requirement on the quality of the distance is not as strict as for the Mantel test. Therefore, the results of Word2Vec and TFIDF using Spearman's quantifier are not significantly different.

The trading volume (VOL) showed the highest performance of all the combinations. The implication fits with news affecting the market from two aspects, price and trading volume, and also confirms our previous findings in Qu et al. (2016).

4.7 Conclusions

Significantly, the proposed framework is able to quantify the relationship between financial news and time series without manually modelling the complexity in the conventional way.

The most remarkable result to emerge from the evaluation is that the simplest strategy based on LOO outperforms the strong baseline, BS. Firstly, this confirms that financial news does affect the market. Secondly, since we use non-overlapping sliding windows for the news

and time series, this also provides evidence that the impact of the news lasts longer than a day.

Generally speaking, the proposed framework has vast potential for practical application. The LOO implemented in the evaluation extracts the potential winner by ranking the change in correlations. We have demonstrated that there are linear correlations between the change and return in the future, and by developing a proper asset allocation algorithm, it is also possible to achieve a better portfolio performance.

There are still many hyper-parameters left to explore, for example, the size of sliding windows for news and time series, the distance metric, the relation quantifier.

We are aware that this research is conducted on a daily timescale, which may prevent us from discovering shorter periods of relation between news and time series. A further study using hourly or even minute-sliced data remains an interesting direction for extending the existing research.

Chapter 5

Learning Optimal Weights of Text Features for Financial Forecasting

5.1 Introduction

The amount of available financial news has dramatically increased over the past few years. It is the most important source of information when investors need to carry out fundamental analysis. Although natural language processing techniques are capable of extracting structured information, such as named entities, sentiments or relationships from the news, feature selection for financial news is still part of ongoing research.

Based on the literature, there are three approaches to feature selection. The first is dictionary-based, which requires a huge amount of effort from domain experts from finance and other related fields (Loughran and McDonald, 2011; Thomas and Routledge, 2003). The second approach is conventional feature selection in text classification approaches, e.g. TFIDF . This does not take market feedback into account. The third approach employs exogenous market feedback, e.g. Mittermayer and Knolmayer (2006), which selects the most relevant features, so that positive and negative news can be discriminated.

However, market feedback contains richer information than simply a positive or negative label on the text. The framework proposed in Chapter 4 inspired us to improve feature se-

lection with a better way of exploiting market feedback. Briefly speaking, this novel approach uses the framework in Chapter 4 and determines the weights of features by maximising the quantified relationship.

This chapter is organised as follows: Section 5.2 outlines our methodology for quantifying the relationship between news and the time series based on the Mantel test. In Section 5.3 the data and experiment are presented. Section 5.4 shows the results obtained by our proposed framework as well as an evaluation and comparison between our proposed framework and common baselines. Conclusions are drawn in the final Section 5.5.

5.2 Methodology: Feature Selection by Optimising Quantified Relation

We first briefly revisit the framework we proposed in Chapter 4. The key idea of the framework is to measure two kinds of distance between two stocks: distance between textual information and distance between time series, and then quantify the relationship between the two kinds of distance.

In Chapter 4, we passively measure the distances and quantify the correlation. The effectiveness of the quantified relationship is proven to select market winners. Conversely, we are able to assign and tune representation of texts and actively make the quantified relationship with a higher value, such that we can identify which textual features are useful by their corresponding weights. Figure 5.1 illustrates the idea of our approach.

The routine of our approach consists of the following steps which are similar to the procedure explained in Chapter 4:

1. Choose textual feature and representation, assign weights which are the variables to be learned during optimisation.

2. Choose distance aggregation function between text representations

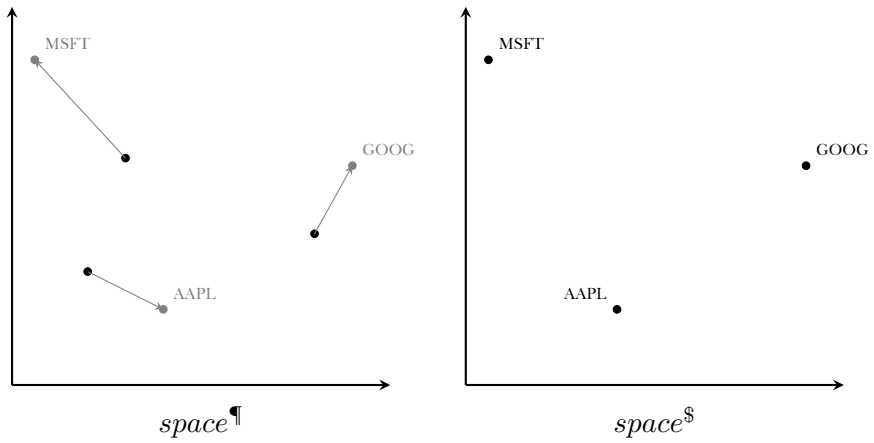


Figure 5.1: Illustration of our approach. Grey dots represent vectors of stocks after optimisation. In which case, the pair-wised distances (AAPL-MSFT, AAPL-GOOG, etc.) will produce higher quantified relationship than those without optimisation (black dots).

3. Set the target to be optimised; it can be either the Pearson's correlation or the Mean Squared Error between textual distances and time series distances.

4. Optimise the weights

5.2.1 Optimise Based on Word2Vec and the Pearson's Correlation Coefficient

We demonstrate our approach by the Word2Vec representation. Consider N_{stock} stocks in our portfolio at time t sliding window. Each stock has its own news articles out of the overall document set $\mathbb{D} = \{\boldsymbol{d}_1, \cdots, \boldsymbol{d}_{N_{news}}\}$. In our Word2Vec model, an individual document is represented by N_{dim} dimension vector. In Chapter 4, this document vector is obtained by averaging over all the word vectors in it. Here, differently, we assign each word a weight such that:

$$\boldsymbol{d} = \frac{\sum_{v_i \in d} w_i \boldsymbol{v}_i}{||d||} \tag{5.1}$$

Our task is to learn the weights so that quantified relationship, i.e. the Pearson's Correlation Coefficient between all pair-wised $distance^{¶}$ and $distance^{\$}$ can be maximised. Here, the Pearson's Correlation Coefficient is chosen as the target because originally, in Chapter 4, we used the Mantel test which relies on the Pearson's Correlation Coefficient as the original Mantel

correlation ρ_0 defined in 4.7, and a higher ρ_0 improves the probability of getting statistically significant Mantel correlation. On the other hand, our optimisation requires a differentiable loss function such that optimal weight can be obtained.

From Equation 4.7, we define the loss function as follows:

$$loss = -\rho_0 = -cosine(\boldsymbol{x} - \bar{\boldsymbol{x}}, \boldsymbol{y} - \bar{\boldsymbol{y}}) \tag{5.2}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the **upper triangle elements** in the same rank from distance matrix $\boldsymbol{M}^{¶}$ and $\boldsymbol{M}^{\$}$

$$\boldsymbol{x} = \langle m^{¶}_{1,2}, m^{¶}_{1,3}, \cdots, m^{¶}_{2,3}, \cdots \rangle \tag{5.3}$$

$$\boldsymbol{y} = \langle m^{\$}_{1,2}, m^{\$}_{1,3}, \cdots, m^{\$}_{2,3}, \cdots \rangle \tag{5.4}$$

where $m_{i,j}$ represents a pair-wised distance between $stock_i$ and $stock_j$. The distance between the two stocks i and j in $space^{¶}$ is calculated by

$$m^{¶}_{i,j} = \texttt{distance}^{¶}(\boldsymbol{D}_i, \boldsymbol{D}_j) \tag{5.5}$$

$$= WMD(\boldsymbol{D}_i, \boldsymbol{D}_j) \tag{5.6}$$

where $\boldsymbol{D}_i$ is the matrix representation of the set of news for stock i, $N^{(i)}_{news} \times N_{dim}$ dimensional. Here, we use the aforementioned WMD to obtain textual distances. According to Chapter 4, we choose the trading volume as the time series and use DTW to get corresponding $m^{\$}_{i,j}$ in $\boldsymbol{M}^{\$}$, and $\boldsymbol{M}^{\$}$ can be pre-computed.

5.2.2 Learning Optimal Weights

We use the *Gradient Descent* algorithm to obtain the optimal weights. First we define a loss function:

$$J(\boldsymbol{W}) = -cosine(\boldsymbol{x} - \bar{\boldsymbol{x}}, \boldsymbol{y} - \bar{\boldsymbol{x}}) \tag{5.7}$$

The approximate process is repeating the following step to minimise the loss until a condition is satisfied (see 1 for detail):

$$\boldsymbol{W} := \boldsymbol{W} - \eta \nabla J(\boldsymbol{W}) \tag{5.8}$$

where η is learning rate, and $\nabla J(\boldsymbol{W})$ is gradient of J.

5.2.3 Baseline Approach: Word2Vec Embeddings with TFIDF

Considering our approach is to learn optimal weights for each word in the vocabulary and obtain the sentence vector by the weighted average vector, we choose a similar approach such that TFIDF weights are applied to each word in a sentence. This approach has been verified in (Zhao et al., 2015; De Boom et al., 2016). In the approach, Equation 5.1 can be written as

$$\boldsymbol{d} = \frac{\sum_{v_i \in d} tfidf_{v_i,d} \boldsymbol{v}_i}{||d||} \tag{5.9}$$

5.3 Experimental Design

5.3.1 Data Description

The data used in this chapter is the same as in Chapter 4, in which news of 25 companies was collected from Yahoo Financial News from 1st October 2014 to 30th April 2015.

All texts are processed in the same way as described in chapter 4 and the same BoW model with TFIDF weight scheme is used and arrived at 14451 unique tokens. Considering standardised *volume* (VOL) provides the best Mantel correlation over all periods according to the results of Chapter 4, and we use it as the financial time series in our experiments. For the purpose of comparison with previous results in Chapter 4, we use the same sliding window

Table 5.1: Dataset split for experiment. Note that dates are inclusive on the left and exclusive on the right.

Dataset	Sliding Window Indices	News Items	Samples
Training	1 - 63 (3 months)	26757	64
Validation	64 - 103 (2 months)	16265	40
Test	104 - 142 (2 months)	21681	39

settings: 7 calendar days sliding window for the news and 5 trading days sliding window for the time series.

Table 5.1 shows that the whole dataset is split into three subsets: the first three months' data, from 1st October 2014 to 31st December 2015, is used as a training dataset to perform the optimisation process. Data in the next two months, from 1st January 2015 to 28th February 2015, is used as a validation set to determine if the optimisation is over-fitting with the training data. The last two months' data, from 1st March 2015 to 30th April 2015 is used as the test dataset to see how the optimised weights of features can improve the Mantel correlation.

5.3.2 Optimising

For the *Gradient Descent* optimisation, we use the *Early-Stopping* strategy to avoid over-fitting, and the patience is set to 100. On the other hand, from practical numerical computation, to avoid accuracy problems we set an upper threshold $\epsilon_b = 10^{-6}$ and $\epsilon_u = 10^3$. Once the condition $\exists \boldsymbol{W} \leq \epsilon_b$ or $\exists \boldsymbol{W} \geq \epsilon_u$ is satisfied, the *Gradient Descent* will be stopped.

5.3.3 Evaluation

The evaluation follows the same *Leave-One-Out* procedure described in Section 4.5.3, however we can only perform the evaluation on the sliding windows from the test range, not on all sliding windows from the beginning. We pick up the best strategy obtained using the Mantel correlation and volume as time series, namely the Word2Vec representation. For all sliding windows in the test range, we split the strategy into three branches:

Word2Vec : continue without any change.

Word2Vec +TFIDF : switch text representations to TFIDF weighted Word2Vec .

Word2Vec +$W_{optimal}$: switch text representations to Word2Vec with learned optimal weights.

The **BS** and **UCR** (see 4.5.3) strategies are also kept in our evaluation for the purpose of comparison.

5.4 Results and Discussion

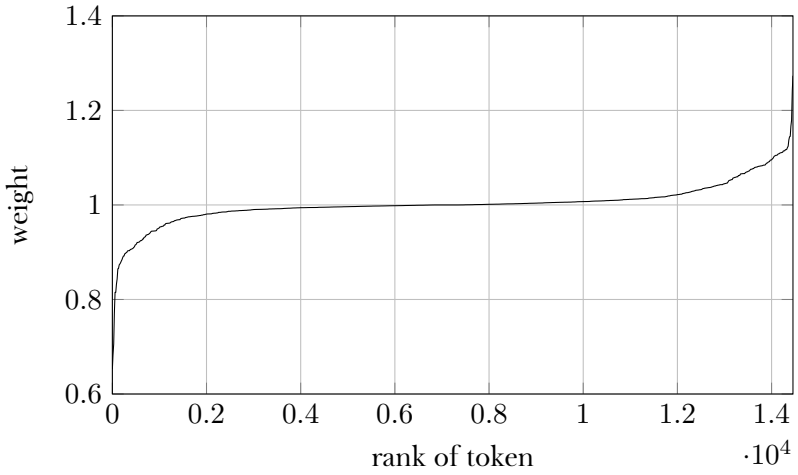


Figure 5.2: Distribution of optimised weights of unique words, ranked from low to high.

Figure 5.2 shows the distribution of optimal weights obtained. We ranked the weights from lowest to highest and the figure shows that most words keep a weight around 1.0, with a few words optimised to have weights that significantly differ from 1.0, where the maximum reaches above 1.4 and the lowest reaches below 0.7. Surprisingly, this result is very similar to plots in the literature (see Junqué De Fortuny et al., 2014, Fig.6). Table 5.2 shows sampled terms from the top, middle and bottom positions of ranking based on optimal weights. We also found that higher weighted and lower weighted terms are mostly component words of named entities (see Junqué De Fortuny et al., 2014, Table.10).

Table 5.2: Sampled terms from top, middle and bottom in the ranking with regards to optimal weights.

From	Terms
Top 100	tesla, musk, model, car, starbucks, cisco, food, paypal, motors, ebay
Middle 100	count, instead, weather, la, little, daily, fly, hot, drone, wait
Bottom 100	dow, amazon, lte, product, ibm, ceo, china, earnings, fcc, growth

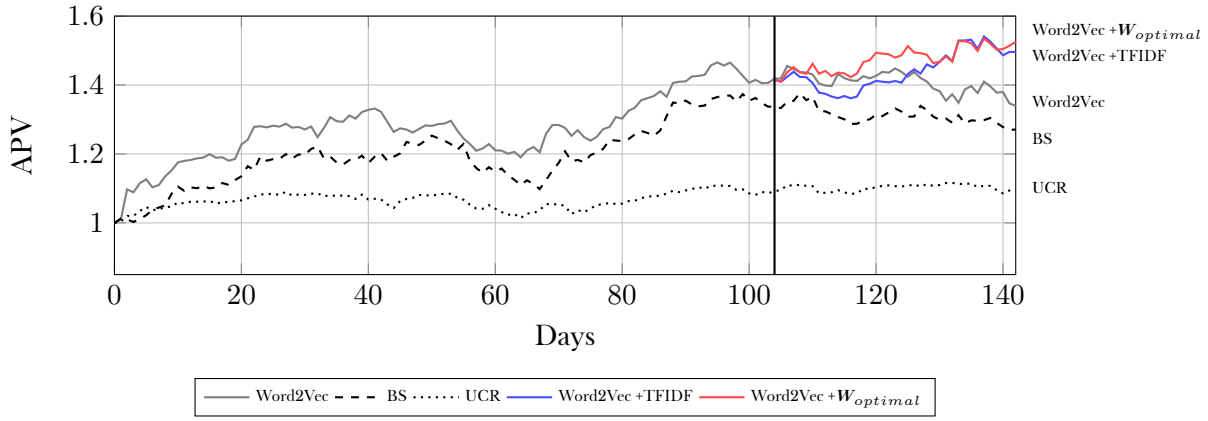


Figure 5.3: Accumulative Portfolio Values (APV) of strategies. The curves after the 103rd day (black vertical line) are in the test range in which the Word2Vec splits into three branches: Word2Vec , Word2Vec +TFIDF , Word2Vec +$\boldsymbol{W}_{optimal}$.

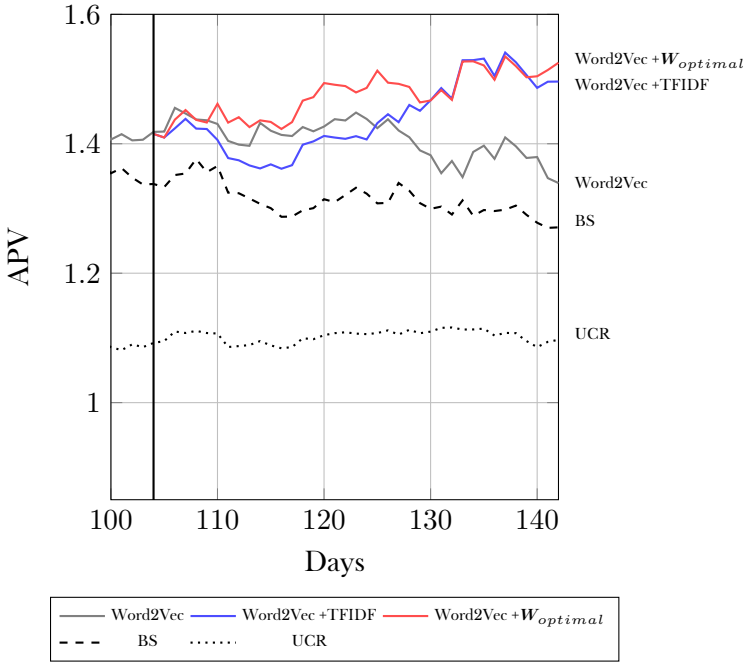


Figure 5.4: Accumulative Portfolio Values (APV) of strategies (zoomed in). The curves after the 103rd day (black vertical line) are in the test range in which the Word2Vec splits into three branches: Word2Vec , Word2Vec +TFIDF , Word2Vec +$\boldsymbol{W}_{optimal}$.

Figures 5.3 and 5.4 show the accumulative portfolio values after switching to different representations: Word2Vec +TFIDF and Word2Vec +$\boldsymbol{W}_{optimal}$. The figures show that the output of both representations perform the Word2Vec without a weighting scheme on the final cumulative value.

Table 5.3: Financial Performances of trading strategies during test date ranges. There are three groups of participants. The first group shows results using normal Word2Vec for text representation; the second group shows results using TFIDF and learned optimal weights; in the last group BS and UCR are baselines, where BS is the Best Stock strategy, and UCR is the Uniform-Buy-and-Hold strategy. fAPV refers to final-Accumulated Portfolio Value; SR stands for Sharpe Ratio; MD is Maximum Drawdown; AR is the Annualised Return. Bold numbers are the best participants.

Representation	**fAPV**	**SR**	**MD [%]**	**AR [%]**
Word2Vec	0.952	-1.444	8.980	27.014
Word2Vec +TFIDF	1.064	2.137	5.358	49.239
Word2Vec +$\boldsymbol{W}_{optimal}$	**1.085**	**2.695**	3.245	**69.009**
BS	0.950	-1.876	7.730	-27.997
UCR	1.010	0.736	**2.713**	6.913

Table 5.3 shows the strategy performances using fAPV, SR, MD and AR (details in Section 4.5.3). Note that the result is obtained using only the data from the test period, which was from the 103rd day to the end, a total of 39 trading days. Therefore, fAPV may not be a significant metric for comparison. Word2Vec +$\boldsymbol{W}_{optimal}$ shows the best performance of the three metrics, apart from the maximum drawdown, but it was very close to the lowest UCR strategy at 3.245%. Significant improvements were made by TFIDF and $\boldsymbol{W}_{optimal}$ compared with the normal Word2Vec representation, which reveals that our approach is effective.

5.5 Conclusion

We have described here a novel feature selection technology, which under the framework we proposed in Chapter 4, is able to assign optimal weights to individual words, such that news and time series can be quantified with a higher value. The results show that by improving text representation, the performance of our proposed trading strategy (described in 4.5.3) can be

greatly improved.

This also indirectly confirms the effectiveness of the framework proposed in Chapter 4. Interestingly, the finding that named entity-related words tend to be assigned with weights which differ more from 1.0, provides supportive evidence to the assumption that named entities are critical features in financial forecasting.

We are aware that our research may have limitations due to the amount of data we used. This limitation underlines the difficulty of collecting data pertinent to financial forecasting using text. This could potentially affect the robustness of the outcome of the research. Therefore, future studies should examine the robustness using more data.

Apart from the limitations, our results are encouraging and give us confidence to extend and explore more possibilities of varying the components of the framework we proposed in Chapter 4.

Chapter 6

Financial Forecasting Based on Stock Charts

6.1 Introduction

Financial forecasting means making predictions, based on historical information (usually daily data), of the future price direction of assets (up or down, sometimes neutral), also known as directional forecasting.

Representation is an important task in financial forecasting. Many time series representation techniques have been proposed. However, financial time series have unique characteristics. The most obvious thing is that it is usually not a single time series but a lower sampled sequence of data consisting of open price, high price, low price, close price and volume of trade. Almost all existing representation techniques are only able to handle one time series at a time. On the other hand, financial time series are fuzzy and non-stationary compared with other types of time series (e.g. temperature or traffic). Very little research has been conducted specifically on representing financial time series.

Candlestick charts, one of the most popular data visualisation techniques in the financial industry, have been shown to be successful in providing assistive visual patterns such as for trend and duration. Visual patterns are very widely used in financial technical analysis and

their importance has been studied in the literature reviewed in Ko et al. (2016). A number of studies have investigated the potential usage of visual patterns for making better financial forecasts, for example, Sandoval and Hernández (2015) and Sandoval et al. (2016) present studies which show that visual patterns of price-time-volume help in predicting the future movement of the foreign exchange market. Lee et al. (2006) explore the idea of extracting patterns from candlestick charts to assist financial prediction. Leigh et al. (2008) conduct systematic research and present statistically significant results to show that trading price history patterns can provide predictive signals to help make a higher return than with a random strategy.

Our motivation is to learn from stock charts, like human traders, and make financial forecasts. Deep learning techniques proposed in recent years have utilised successful state-of-the-art approaches in the field of image recognition tasks, which in our opinion have great potential for solving our problem. Deep learning has been introduced in the field of financial forecasting but without using vision representation techniques, as we have done (Hu et al., 2017; Chen et al., 2015; Chong et al., 2017, inter alia).

This chapter is organised as follows: Section 6.2 gives an introduction to related work in the field of image recognition and deep neural networks. Section 6.3 presents the details of the approach by explaining how samples for training and testing the neural network are generated, as well as competitive network structures. Section 6.4 shows the results of the experiment with a discussion. Section 6.5 summarises the chapter and draws conclusions.

6.2 Related Work

Batch Normalisation

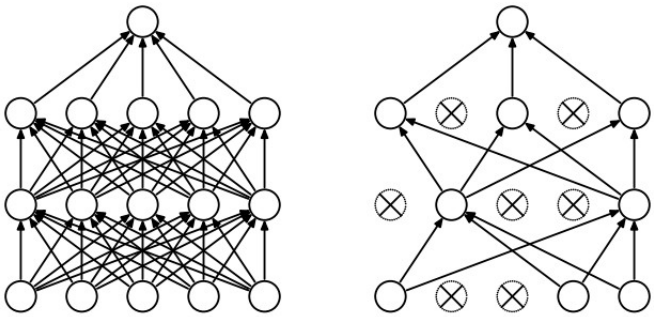
Batch normalisation is a technique that facilitates faster deep neural network training by normalising over each dimension of a mini-batch and applying scale and shift operations to a layers' output.

$$\hat{\boldsymbol{x}} = \frac{\boldsymbol{x} - \mu}{\sigma} y = \gamma \hat{\boldsymbol{x}} + \beta \tag{6.1}$$

where μ is the mean of $\boldsymbol{x}$ in a mini-batch, σ is the standard deviation of $\boldsymbol{x}$, γ is the scale factor and β is the shift factor (both are parameters to be learnt during training).

Dropout

Dropout (Srivastava et al., 2014) is a technique to avoid the over-fitting problem when training a neural work. Figure 6.1 shows the idea of dropout.



(a) Standard Neural Net (b) After applying dropout.

Figure 6.1: Illustration of dropout mechanism from the original dropout paper (Srivastava et al., 2014). The left figure shows a standard neural network with two hidden layers. The figure on the right shows the network with applied dropout technique during training. Crossed units are the dropped out nodes.

Consider a hidden layer with weights $\boldsymbol{W}$ and biases $\boldsymbol{b}$, when dropout applies to the layer

$$r_j \sim Bernoulli(p) \tag{6.2}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{r} \odot \boldsymbol{y} \tag{6.3}$$

$$\hat{h_i} = \boldsymbol{w}_i \hat{\boldsymbol{x}} + b_i \tag{6.4}$$

$$\hat{o_i} = F(\hat{h_i}) \tag{6.5}$$

$$h_i = p\boldsymbol{w}_i \boldsymbol{x} + b_i \tag{6.6}$$

$$o_i = F(h_i) \tag{6.7}$$

where $\odot$ is an element-wise product, $\boldsymbol{r}$ is a vector of Bernoulli random variables in which each element has probability $1-p$ to be 0 (dropped) and p to be 1 (kept). $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{o}}$ are the input and output vector during training, $\boldsymbol{x}$ and $\boldsymbol{o}$ are for test. Note than weights in the test stage are scaled by p. In other words, dropout randomly activates a portion of p units of the layer for each training batch, and this is equivalent to training subnets ("thinned" network) of the neural network, and the final neural network is averaged from all its subnets.

6.2.1 Image Classification based on Deep Neural Networks

Our problem can be categorised as an image classification task, i.e. given an image (stock chart), we can predict which class (degree of price change) it belongs to. The use of Deep convolutional networks (DNN) has been successful in this field. The basic paradigm of DNN to solve the image classification problem is to extract features of different sizes and locations from input images using a convolutional layer, then abstract the extracted features by stacking up more layers.

Fully Convolutional Networks

The fully convolutional network (FCN) structure was proposed by (Long et al., 2015) to improve whole-image classification tasks. The idea of FCN is to stack convolutional blocks (convolution, pooling, activation, etc.) so that the locations in higher layers corresponding to the locations in the input are "path-connected" to each other (so-called receptive fields). The FCN contains layers that

$$y_{ij} = f_{ks}(\{\boldsymbol{x}_{s_i+\delta_i, s_j+\delta_j}\}_{0\leq\delta_i,\delta_j\leq k}) \tag{6.8}$$

$$f_{ks} \circ g_{k's'} = (f \circ g)_{k'+(k-1)_{s',ss'}} \tag{6.9}$$

where k is the kernel size, s is the stride, and f_{ks} is the forward function of the layer. The kernel size k and s are constrained by the above equations so that receptive fields can overlap with each

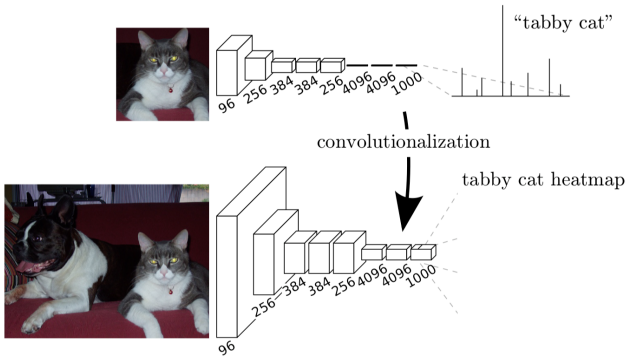


Figure 6.2: Figure from (Long et al., 2015). It illustrates fully connected layers. The cat picture above represents the receptive field in the input image.

other, leading to more efficient computation layer-by-layer over the whole input image in both the feedforward and back propagation stage. This enables the network to produce heatmaps layer by layer which can be used for pixel-wise prediction or classification (see Figure 6.2).

Deep Residual Neural Network

The number of layers is proven to be a critical hyper-parameter. However, a network with a lot of layers suffers from vanishing or exploding gradients when it needs to be trained. (He et al., 2015) proposed Residual Neural Network by identity mapping which is defined as

$$y = F(x, \{W_i\}) + x \tag{6.10}$$

where F is a projection function of a neural layer, x is the input vector and y is the output vector. The most promising advantage is that residual networks greatly overcome the vanishing or exploding gradients problem, even with a lot of layers.

6.3 Methodology

6.3.1 Preparing Samples

Generating Stock Charts

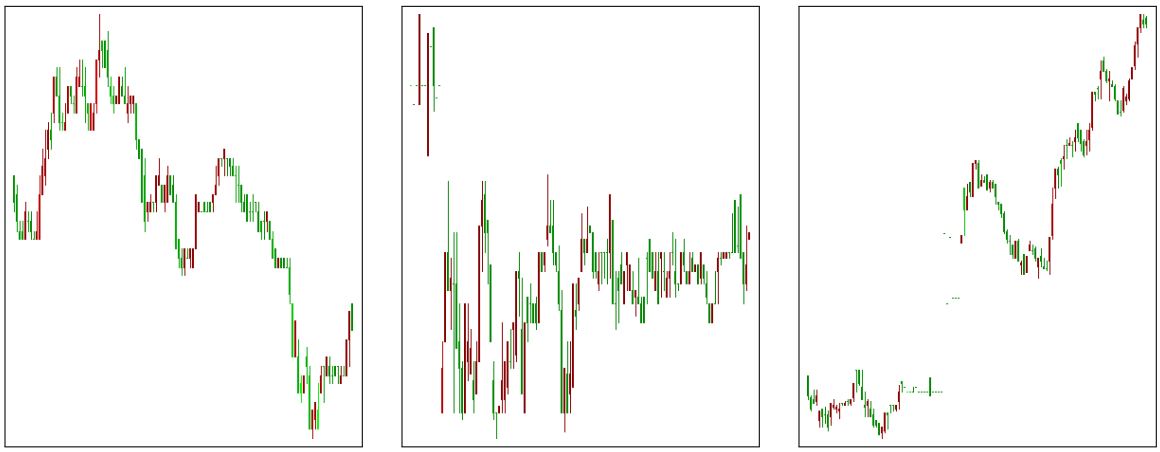
Traders use stock charts to understand stock data as they are intuitive and it is easy to discover patterns in them. Candlestick charts are the most widely used charts when visualising stock prices. In a candlestick chart, each bar represents a time frame (see Figure 2.1), for example, if we choose a day as the time frame, we get a daily candlestick chart. The candlestick consists of a box and a pin, where the bottom and top of the box are determined by the stock price at the beginning (the open price) of the time frame and at the end (the close price). The top and bottom of the pin are determined by the highest (high price) and lowest prices (low price) during the time frame. In order to distinguish whether the close price is higher than the open price, the candlesticks are displayed using two different colours, usually green for up and red for down.

Figure 6.3 shows three training samples. Each candlestick chart is obtained from a sliding window at t over a period of time. The candlestick chart used in this research has a height of 224 RGB[1] pixels and the width of each candle bar is 2 RGB pixels, plus 1 RGB pixel between the candlesticks. In other words, a candlestick chart is actually a tensor with the shape of $(3 \times s, height, 3)$, where s is the size of the sliding window.

Scaled Time Series

Although a candlestick chart is an effective data visualisation method for human traders to interpret and analyse stock prices, the representation is redundant compared with the amount of data it encodes. Therefore, we are also interested in directly using a financial time series

[1]RGB refers to Red Green Blue, an RGB pixel is an array of three values. The value indicates the degree of corresponding colour, usually between 0 and 255.



(a) label=DOWN (b) label=NEUTRAL (c) label=UP

Figure 6.3: Examples of candlestick charts as training data, generated from IBM minute level OHLCV.

as input. The only difference is how we represent the data in a single sliding window at time t. Given OHLCV, a clip of trading data, we scale each time series, open, high, low, close and volume between 0 and 1, respectively using the following equation:

$$scale(\{x_t\}) = \{\frac{x_i - min(\{x_t\}))}{max(\{x_t\}) - min(\{x_t\})}\}, i \in \{1, 2, \cdots, N\} \tag{6.11}$$

where N is the length of time series $\{x_t\}$. Figure 6.4 shows three samples of generated input images and their labels.

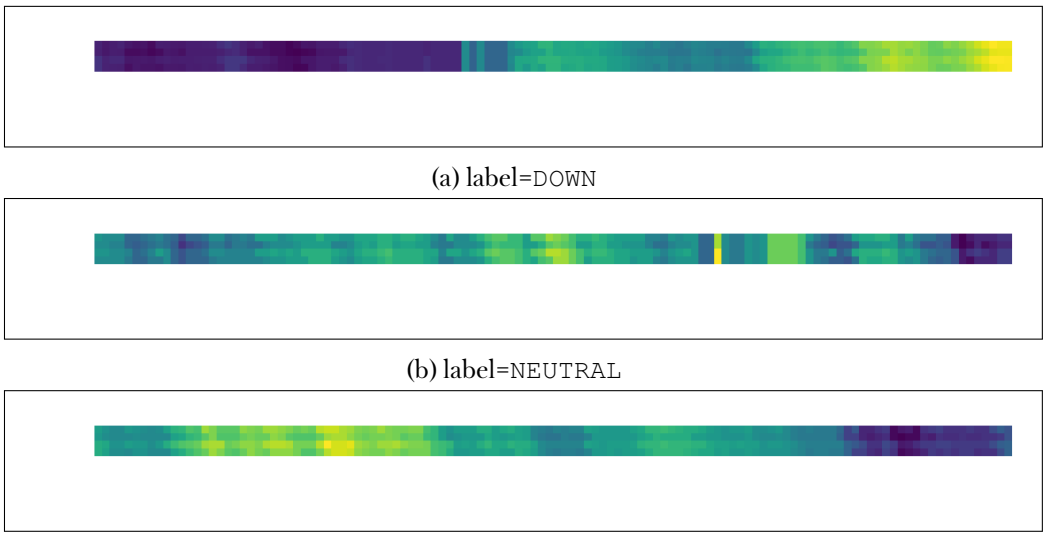
Labelling Samples

The label is produced according to $return_{t+1}$ next to the sliding window.

$$label_t = \begin{cases} \texttt{UP}, & \text{if } return_{t+1} > \theta \\ \texttt{DOWN}, & \text{if } return_{t+1} < -\theta \\ \texttt{NEUTRAL}, & \text{otherwise} \end{cases}$$

where

$$return_{t+1} = \frac{close_{t+1} - close_t}{close_t} \tag{6.12}$$



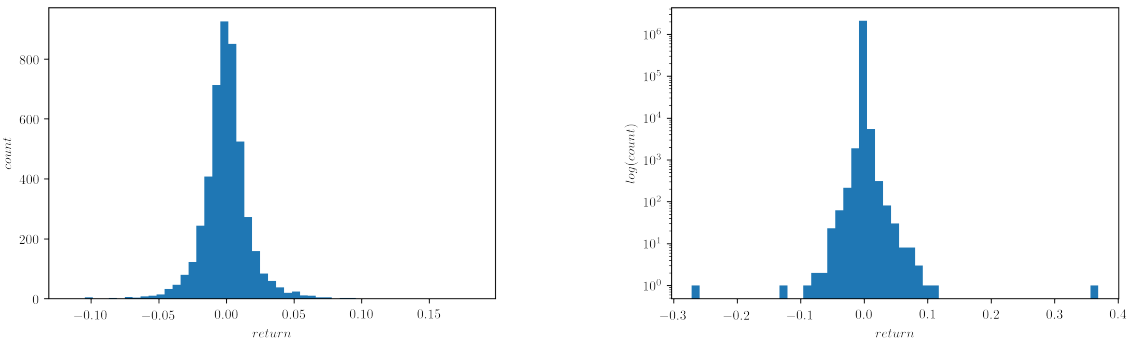
(a) label=DOWN

(b) label=NEUTRAL

(c) label=UP

Figure 6.4: Examples of scaled trading data as training data, generated from IBM minute level OHLCV. Darker pixels represent higher values.

θ is the threshold to determine if *return* is NEUTRAL or not. Daily OHLCV and minute OHLCV have different distributions of *return*, Figure 6.5 shows histograms of *return*. Since *return* is approximately a stationary time series, we choose fixed threshold θ over the whole time. For daily data, we use $\theta = 0.5\%$ and for minute data $\theta = 0.015\%$, so that each label has an equal portion of samples.



(a) Daily

(b) Minute-by-minute

Figure 6.5: Histogram plot of *return* of daily and minute OHLCV data of IBM stock.

Table 6.1: Sample distribution of the training data.

Interval	θ	Label			Total
		UP	Down	Neutral	
Day(IBM)	0.500%	1465 (32.347%)	1557 (34.378%)	1507 (33.274%)	4529
Minute(IBM)	0.015%	650964 (34.775%)	648200 (34.627%)	572785 (30.598%)	1871949

6.3.2 Network Topologies

Two convolutional networks were built. One uses 1D convolutional layers to extract patterns from time series data, the other uses 2D convolutional layers to extract visual patterns from stock bar-charts.

Fully Convolutional Network 1D (FCN1D)

FCN1D is a network with a convolutional layer that performs one dimensional convolution operation directly on a time series. Each convolutional block consists of a 1D convolutional layer, a batch normalisation layer and a dropout layer. The batch normalisation layer is used for a faster convergence for optimisation and the dropout layer is used for the purpose of avoiding over-fitting.

Fully Convolutional Network 2D

FCN2D is a network with a convolutional layer that performs 2D convolution operation on candlestick charts. In addition to a convolutional block similar to FCN1D, we apply Max Pooling after each convolutional layer to reduce the number of parameters.

ResNet2D

ResNet2D has the same structure as ResNet50 6.8 but with a different number of classification labels. The original ResNet50 is an ImageNet used for an image classification task that predict labels for 1000 different objects. In our approach, we tailor the network by replacing the final classification layer with a directional classification layer, which is a fully connected layer with

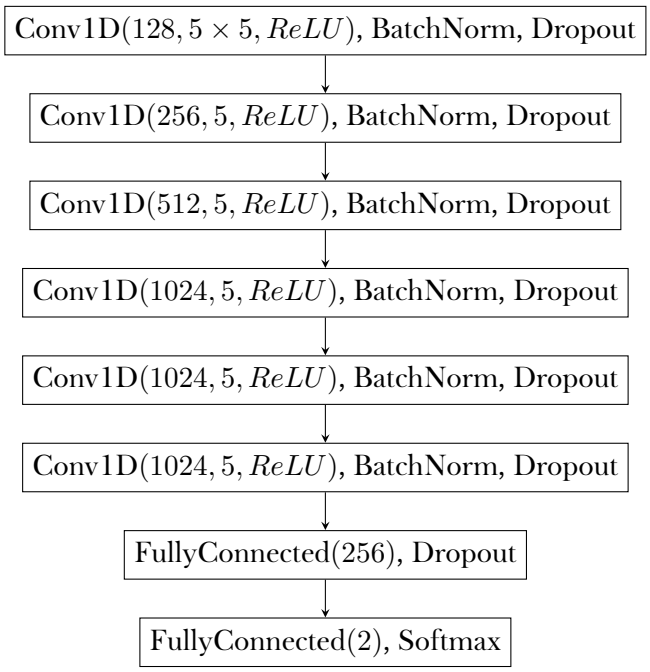


Figure 6.6: Structure of FCN1D network. Conv1D refers to a one dimensional (1D) convolutional layer.

tree units and a softmax activation function.

6.3.3 Loss Function

Our model needs to produce predictions that approximate the ground-truth, which are binary matrices. This is equivalent to a multi-labeled classification problem in which predictions are probabilities. We use *Cross Entropy* (CE) to measure the error between a prediction and the true classes.

$$CE(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N}\sum^{N}(1-\mathbf{y}) \odot log(1-\hat{\mathbf{y}}) + \mathbf{y} \odot log(\hat{\mathbf{y}}) \tag{6.13}$$

where $\mathbf{y}$ is the ground-truth labels, $\hat{\mathbf{y}}$ is the predictions, N is the total number of elements in predictions, and the symbol $\odot$ refers to element-wise multiplication.

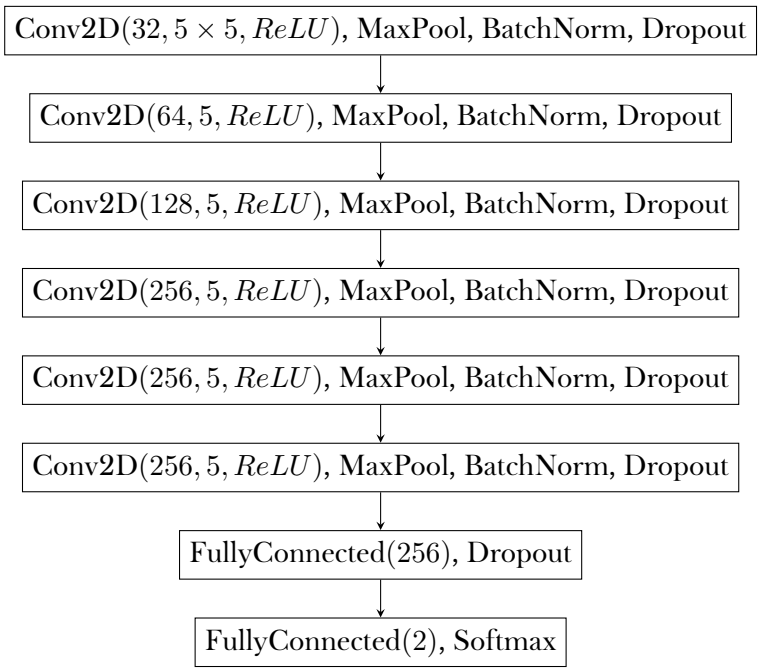


Figure 6.7: Structure of FCN2D network. Conv2D refers to 2D convolutional layer.

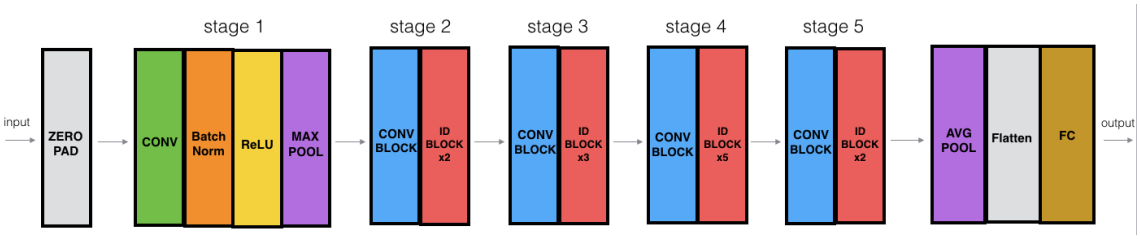


Figure 6.8: Illustration of ResNet50 structure (Malhotra, 2017) by stages.

6.3.4 Experimental Design

Data Description

In order to examine how our model performs under different time intervals, we use daily level trading data and minute level trading data of IBM stock. Considering there are no trades after market hours, the sliding window only slides within trading hours of the same day.

Table 6.2: Details of training, validation and test dataset used (all from IBM stock).

Dataset	Training	Validation	Test
From	1998-01-02 09:30	2016-01-04 04:05	2017-01-03 04:00
To	2015-12-31 18:15	2016-12-30 18:59	2017-12-29 19:04
Minutely (rows)	1871949	102639	103231
Daily (rows)	4529	252	250

Training the Network

The training strategy we use is *Early Stopping* (ES) for its proven superior performance (Fiesler, 1994). The strategy examines the performance of the model on the validation set at the end of each training epoch and updates the snapshot of the model as the best one when a better validation result is detected. The procedure repeats but not forever; if after a number of epochs, so-called patience, when there is no update of the best model, the training is considered sufficient and the best model obtained is considered ready. We use the patience of 50, which means if after 10 epochs the obtained validation loss is still higher than the lowest validation loss, the training process will be terminated. Table 6.3 shows the hyper-parameters used in the training process.

Table 6.3: Hyper-parameters for Network Training

Hyper-parameter	**value**
Max. Epoch	50000
Bath Size	10
Patience	100
Dropout ratio	0.5
Optimiser	Adadelta
Learning Rate	0.001
Input Sliding Window	70 Days
Forecast	1 Day Ahead

6.4 Results and Discussion

Table 6.4 shows the result obtained using minute-by-minute `OHLCV` data. Table 6.4 shows the result obtained using daily `OHLCV` data.

Table 6.4: Result of performances of FCN1D, FCN2D, ResNet50 on **minute interval** stock data. All evaluated using test samples. DA refers to Directional Accuracy, P for Precision, R for Recall, F1 for F1-Score and Support is number of samples with the specified label.

Model	DA[%]	Label	P	R	F1	Support
FCN1D	40.82	UP	0.26	0.23	0.24	26674 (26.023%)
		DOWN	0.26	0.11	0.15	26465 (25.820%)
		NEUTRAL	0.48	0.66	0.56	49361 (48.157%)
		avg	0.37	0.41	0.37	102500
FCN2D	38.41	UP	0.32	0.21	0.26	Same as above
		DOWN	0.29	0.11	0.15	
		NEUTRAL	0.42	0.70	0.52	
		avg	0.35	0.38	0.34	
ResNet50	**43.78**	UP	0.34	0.14	0.20	
		DOWN	0.28	0.50	0.32	
		NEUTRAL	0.34	0.14	0.20	
		avg	0.32	0.31	0.29	

Table 6.5: Result of performances of FCN1D, FCN2D, ResNet50 on **minute interval** stock data. All evaluated using test samples. DA refers to Directional Accuracy, P for Precision, R for Recall, F1 for F1-Score and Support is number of samples with the specified label.

Model	DA[%]	Label	P	R	F1	Support
FCN1D	38.55	UP	0.36	0.11	0.16	38 (21.23%)
		DOWN	0.21	0.37	0.26	41 (22.91%)
		NEUTRAL	0.53	0.50	0.51	100 (55.87%)
		avg	0.37	0.41	0.37	179
FCN2D	26.82	UP	0.33	0.26	0.29	Same as above
		DOWN	0.24	0.83	0.37	
		NEUTRAL	0.57	0.04	0.07	
		avg	0.35	0.38	0.34	
ResNet50	**43.57**	UP	0.19	0.08	0.11	
		DOWN	0.28	0.54	0.36	
		NEUTRAL	0.64	0.53	0.58	
		avg	0.51	0.40	0.42	

6.5 Conclusion

We have shown that ResNet50 outperforms both FCN1D and FCN2D with higher directional accuracy by using stock charts as input. However, FCN2D with the stock chart as input gives lower performance than FCN1D using the scaled time series. On the other hand, we observe a lot of low precision and recalls, especially for those samples with UP and DOWN.

The performance of ResNet50 confirms the assumption that stock charts contain predictive

power for financial forecasting. On the other hand, although FCN2D also takes stock charts as its input, the performance is rather lower than FCN1D which only uses scaled time series. This implies that capturing predictive patterns from stock charts requires more complex networks which are able to perform tasks such as image recognition.

Regarding the time interval, all networks give better performance on the minute level data than the daily data. This is an unsurprising result considering the limited amount of daily data.

6.5.1 Future Work

Network Structure

Deep learning networks are under heavy exploration from many aspects, especially in terms of network structure. This research can be further extended by using other topologies, such as Recurrent Neural Networks with their proven successful results on temporal dependent tasks. We have tried various network structures beyond the reported ones.

An Alternative Approach Instead of Directional Forecasting

Practically speaking, directional forecasting is an over-simplified approach. Unlike temperature or electricity consumption, a financial time series is not a single time series but a series of status snapshots of an asset or market. Therefore, directional forecasting only provides predictions at the time of snapshots, while volatility between snapshots is unintentionally ignored. The longer the time interval for which directional forecasting is made, the less useful it is. For example, for daily trading frequency, the price actually may swing around a certain range between high and low prices, and the range usually covers both up and down regarding the close price. Directional forecasting lacks information of volatility and may cause risky decisions. Figure 6.9 presents an example to show why the directional approach can lead to bad decisions.

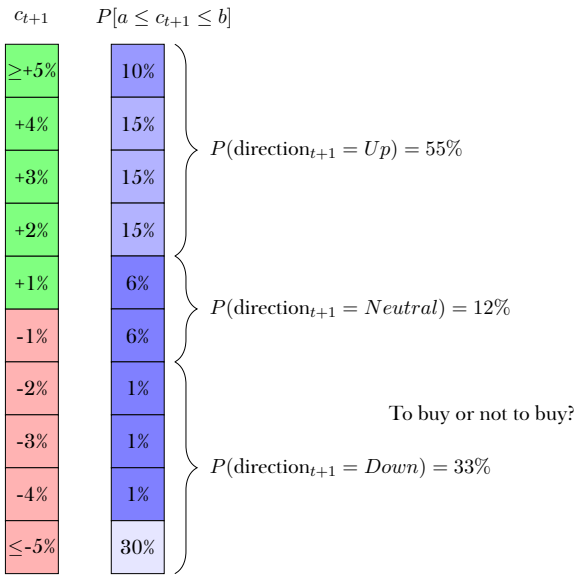


Figure 6.9: To buy or not to buy? This figure shows an example of directional forecasting causing an extremely risky investment decision. Assume a perfect forecasting algorithm gives the probability of $p(c_{t+1})$ price change c in future at time $t+1$. c_t is price change at time t, $P[a \leq c_{t+1} \leq b]$ is the probability that the price will change between level a and b, e.g. $P[3\% \leq c_{t+1} \leq 4\%] = 15\%$. The most likely is that the price will drop 5% or even more. However if using the directional approach, the probability of going up is summed up to 55%, higher than the probability of dropping down of 33%. The detailed information of risk distribution is hidden.

A good alternative approach to overcome the limitation is to forecast the probability distribution of price change in future.

Incorporating Textual Information

The use of textual information is increasingly important for financial forecasting nowadays. Integrating textual information into a time series based forecasting network is a very tempting approach to give potentially positive results. A straightforward way to achieve that is to provide the on-time textual information (Word2Vec for example) as an additional input to existing networks at one or two layers before, but after convolutional blocks. We have explored such an approach, however no positive results were produced.

Chapter 7

Conclusions

7.1 Overview of Findings and Discussion

This thesis presented original work that focuses on financial forecasting using online news.

Most existing studies tackle the problem as a traditional machine learning problem, assigning labels to news and trying to train a learner to predict the direction of future financial movements. After our review of financial forecasting using textual data,it is clear that the presence, effectiveness and predictive power of news remains not very well investigated.

As this research began without a gold standard dataset combining aligned time series and online news, data collection and exploration was conducted first. The result is a valuable dataset that future studies or other researchers can use. Our exploratory study described in Appendix A combining tweets about financial events and financial time series in an ontology style data structure, represents an important contribution offering a template for the integration of such data with broader background information represented in ontological style data. The dataset also offers a starting point for further research into the early detection of the onset of anomalous or catastrophic events, the feared 'black swans' of the stock markets.

The first study was conducted to answer the question about when and how much financial news is related to financial time series? This is an essential issue when the news is intended to be one of the inputs of a forecasting system. Besides, it is also a concern for traders who rely on

news to help them make decisions. With our proposed novel framework, given some stocks as a portfolio, we can answer the question above by measuring their relationship and determining if it is significant. The findings illustrate that news is correlated with different financial time series for some periods of time and the correlation varies. We also illustrated the effectiveness of the quantified relationship by performing a winner selection simulated trading on real-world data. This novel approach has promising potential as it is a flexible framework in which most pipelines in the experimental design can be replaced with alternative techniques.

As our second step, we addressed an important question related to text processing, which has an impact on the performance of a financial forecasting system: What are the useful text features that can help to increase the quantified relationship between news and time series? We transformed the question into an optimisation task aiming to maximise the Mantel correlation by adjusting the weights of text features. The result suggests that when applying learning optimal weights to text features, quantified relationship shows significant improvement on the winner selection indicator by giving better results in a simulated trading.

The final chapter concentrates on whether stock charts can be interpreted by convolutional neural networks and give better results on directional accuracy.

Chapter 3 and appendix A We collected parallel data consisting of financial news and time series which is available upon request for researchers. We also built a dataset for the study of catastrophic financial events that combines news from Twitter with financial time series in an ontological way.

Chapter 4 and appendix B We proposed a novel framework which can be used to detect and quantify the correlation between news and financial time series for a set of stocks. This framework tackles the problem of linking financial news and time series by using a simple yet effective statistical test, the Mantel test. This contribution has been published and received positive feedback from both reviewers and conference attendees.

Chapter 5 Based on our first contribution, we introduced an optimisation based methodology to rank textual features using the framework we proposed. The results imply that named entities are more correlated with financial time series than regular words. We also verified that the optimised weights are transferable for unseen news to increase the Mantel correlation with time series.

Chapter 6 Firstly, stock charts, which are commonly used by human traders, were used as the input instead of raw financial data, so that the ability and advantage of the convolutional neural network can be utilised to process noisy time series and extract useful graphic patterns. Secondly, the idea of providing distribution-like prediction gives better practical usage potential for both human traders, as an indicator, and automatic trading algorithms, as one input.

The thesis introduced a novel way to exploit the content of news about a given company not in isolation, but in comparison with all other financial news available for a given time window. This idea can be compared, on the most general level, to the way inverse document frequency uses the context of all other documents to modify the weights of textual features. At the same time, our idea is clearly different and novel. Furthermore, the ability to single out companies for which there is a significant causal link between news and financial variables can already serve well a trader who needs to focus their attention on prospective trades from a pool of hundreds of stocks. It is possible that the addition of another algorithmic step, such as sentiment analysis, may permit the full automation of the trading process by allowing the algorithm to decide whether to buy or sell the stock singled out by our approach.

7.2 Limitations and Future Work

As we already mentioned in a previous section, the framework is sufficiently flexible to be applied to financial forecasting. So far we have explored the framework and proved its novelty

and significance with simple techniques. Some of them have great potential and need to be further explored.

Our preliminary experiments proved that the chosen representation for news was too simplistic and did not use the full potential of the Word2Vec technique. Two revised versions of this representation were implemented in Chapter 4 (W2Vdep and W2V$words$). A lot of improvements can be explored by incorporating other techniques like sentiment analysis or event embedding.

Alternatively, as implied by the results of Chapter 5, textual features with semantic (named entity) or structural information (syntax subt-ree) encoded can be applied. We are particularly interested in extracting named entity and syntax subtrees and seeing if further improvements can be made.

Appendix A

Integrating Time Series with Social Media Data in an Ontology for the Modelling of Extreme Financial Events

Integrating Time Series with Social Media Data in an Ontology for the Modelling of Extreme Financial Events

Haizhou Qu, Marcelo Sardelich, Nunung Nurul Qomariyah and Dimitar Kazakov
Artificial Intelligence Group, Department of Computer Science, University of York, UK
hq524, msn511, nq516, dimitar.kazakov@york.ac.uk

Abstract

This article describes a novel dataset aiming to provide insight on the relationship between stock market prices and news on social media, such as Twitter. While several financial companies advertise that they use Twitter data in their decision process, it has been hard to demonstrate whether online postings can genuinely affect market prices. By focussing on an extreme financial event that unfolded over several days and had dramatic and lasting consequences we have aimed to provide data for a case study that could address this question. The dataset contains the stock market price of Volkswagen, Ford and the S&P500 index for the period immediately preceding and following the discovery that Volkswagen had found a way to manipulate in its favour the results of pollution tests for their diesel engines. We also include a large number of relevant tweets from this period alongside key phrases extracted from each message with the intention of providing material for subsequent sentiment analysis. All data is represented as a ontology in order to facilitate its handling, and to allow the integration of other relevant information, such as the link between a subsidiary company and its holding or the names of senior management and their links to other companies.

Keywords: Financial forecasting, stock prices, Twitter, ontology

1. Introduction

On 18 Sep 2015, Volkswagen, one of the world's largest and best known automakers, was named by the US Environment Protection Agency (EPA) as being in breach of its regulations concerning the amount of pollution from diesel engines. Volkswagen had manipulated the outcomes of a vehicle emission test by detecting the specific conditions under which the test took place, and adjusting the performance of its diesel engines in other to meet the required pollution targets, while the same vehicle might fail those targets by a vast margin in actual driving conditions.

Several recent models, including Golf, Polo and the Passat equipped with certain diesel engines were confirmed to contain cheating software that would reduce harmful emissions. The revelation led to a fall of more than 30% of the VW stock price in a single day, which continued to fall in the following weeks, as news was gradually released about the number and seniority levels of people who had knowledge of the deception, until the company CEO himself decided to resign and apologise. There was a prolongued period of uncertainty regarding the spread of this deception across the different continents, and the prices continued to tumble as it became clear that it was not limited to the US market. In addition, subsidiary brands, such as Audi, Seat and Škoda soon revealed the existence of similar practices, with a corresponding effect on their own sales figures and share prices.

We observed these events and collected relevant tweets for the period 15–30 Sep 2015, as well as the minute by minute intra-day stock market prices for Volkwagen (stock symbol $VW, as traded on the Frankfurt Stock Exchange), Ford ($F, NYSE) as an example of an automotive company with no links to the scandal, and the same type of data for the S&P500 stock market index (SPY), providing a baseline for comparison with the US economy as a whole. The Twitter data was then enhanced with the addition of extracted key phrases suitable for sentiment analysis, and the entire dataset was stored as an ontology[1].

2. Financial Forecasting

Since the advent of the stock markets, studying and predicting the future of companies and their share price have been the main tasks facing all market participants. It is extremely difficult to achieve an accurate model that remains reliable over time. There is a very famous yet controversial Efficient Market Hypothesis (EMH) (Fama, 1965), which comes in three forms: weak, semi-strong and strong. If the weak form holds true, stock price cannot be predicted using history prices. The semi-strong form of EMH suggests that stock price reveals all publicly available information. The strong form implies that stock prices will always reflect all information including any hidden information, including even insider's information, if the hypothesis holds. Numerous studies show that EMH does not always hold true (Grossman and Stiglitz, 1980; Haugen, 1995; Shleifer, 2000; Shiller, 2003; Butler and Kazakov, 2012). In all cases, attempts to model and forecast the market are based on time series containing the prices of relevant stock along with other relevant information, which often includes indicators of the general state of the market to allow the evaluation of the relative performance of a given company with respect to the general market trends.

3. Mining Twitter

Along with the development of Social Networking, Twitter has became one of the most popular ways for people to publish, share and acquire information. The two characteristics of this service, instantaneity and publicity, make it a good resource for studying the behaviour of large groups of people. Makinf predictions using tweets has proved a popular research topic. Asur and Huberman (2010) used tweet rate

[1] See the data available at `http://j.mp/FinancialEventsOntology`.

time series to forecast movie sales, with the result outperforming the baseline market-based predictor using HSX,[2] the gold standard of this industry. O'Connor et al. (2010) presented a way to use tweets to predict the US presidential polls. The authors concluded that evolution of tweet sentiment is correlated with the results of presidental elections and also with presidential job approval. Tumasjan et al. (2010) used a much smaller dataset of tweets to forecast the result of the 2009 German elections. Eichstaedt et al. (2015) studied the use of sentiment keywords to predict country level heart disease mortality.

Information extraction from social media can be rather challenging, due to the fact that the texts are very short (up to 140 characters only), noisy and written in an informal style, which often contains bad spelling and non-standard abbreviations (Piskorski and Yangarber, 2013).

4. Ontologies For Financial Data

Ontologies are powerful Artificial Intelligence approach to representing structured knowledge. Their use can also facilitate knowledge sharing between software agents or human users (Gruber, 1993). They are often used in text mining to represent domain knowledge, but their use to describe dynamic processes like time series has been much more limited. The use of ontologies has already been considered in the context of Twitter, as well as in the domain of financial news. For instance, Kontopoulos et al. (2013) discuss the benefits of their use when calculating a sentiment score for Twitter data. Mellouli et al. (2010) describe a proposal for an ontology with 31 concepts and 201 attributes for financial headline news. Lupiani-Ruiz et al. (2011) present an ontology based search engine for financial news. Cotfas et al. (2015) have used ontologies to model Twitter sentiments, such as happiness, sadness or affection. Lee and Wu (2015) developed a framework to extract key words from online social messages and update related event ontologies for fast response to unfolding events.

5. The VW Pollution Scandal Dataset

Despite the substantial amout of research on Twitter data in recent years (Bollen et al., 2011; Wolfram, 2010; Zhang et al., 2011; Si et al., 2013), there are very few publicly available datasets for academic research, with some of the previously published datasets becoming unavailable for various reasons. Yang and Leskovec (2011) provide a large Twitter dataset which has 467 million tweets from 20 million users from 1 June to 31 Dec 2009, or 7 months in total, representing an estimated 20–30% of all tweets published during this period. Go et al. (2009) provide a Twitter dataset labelled with sentiment polarity (positive, neutral or negative), and also split into a training set of 1.6 million tweets (0.8 million positive and 0.8 million negative), and a manually selected test set with 182 positive tweets, and 177 negative tweets.

So far, there has not been a publicly available Twitter dataset, which is aligned with company stock prices. We aim to address this gap, with a focus on an extreme financial event, which could prove helpful in revealing the interplay between financial data and news on social media.

[2]Hollywood Stock Exchange

We collected tweets and retweets from 00:00h EDT on 15 Sep 2015 until 23:59h EDT on 30 Sep 2015.[3] In order to retrieve only relevant tweets, we queried the Twitter API using the tags and keywords listed in Table 1.

Table 1: Tags and keywords for the selection of tweets

Tag/keywords	
@vw	#volkswagen
$vow	#volkswagengate
$vlkay	#volkswagencheat
#vw	#volkswagendiesel
#vwgate	#volkswagenscandal
#vwcheat	#dieselgate
#vwdiesel	emission fraud
#vwscandal	emission crisis

One encouraging observation about this dataset is that it contained tweets with relevant information that predated the official EPA announcement that started the VW diesel engine pollution scandal, as shown below.

Published at 2015, September 18, 10:56:35 EDT

> EPA[4] set to make announcement on major automaker $GM $F $TM $FCAU $HMC $NSANY $TSLA $VLKAY $DDAIF $HYMLF http://t.co/02hNHKq9cx

Published at 2015, September 18, 11:47:58 EDT

> .@EPA to make announcement regarding a "major automaker" at 12 noon today. Source says it will involve @VW. No details yet. Stay tuned.

Published at 2015, September 18, 11:51:42 EDT

> Inbox: EPA, California Notify Volkswagen of Clean Air Act Violations

The first and second tweet did not clearly state that Volkswagen was exactly the automaker, the third tweet is the first one with a clear statement which is ahead of EPA official announcement.[5]

A total of 536,705 tweets were extracted. We have chosen the third tweet as a point in time to split the data into the period 'before the news was out', and the one that followed, resulting in 51,921 tweets before11:51:42 on 18 Sep 2015, and 484,784 after that time. Figure 2 shows a histogram of the number of tweets over each 12h period. A brief timeline of relevant events of the Volkswagen scandal according to Kollewe (2015) is listed below:

18 Sep EPA announces that Volkswagen cheated on the vehicle polluiton test. 482,000 VW diesel cars are required to be recalled in the US.

[3]Earlier tweets were also included if they were retweeted during the indicated time interval.

[4]US Environmental Protection Agency

[5]The attentive reader will find it interesting to compare the timing of the EPA announcement with the closing for the weekend of the Frankfurt stock exchange on that Friday.

20 Sep VW orders an external investigation and CEO apologizes to public.

21 Sep Share price drops by 15 billion Euros in minutes after the Frankfurt stock exchange opens.

22 Sep VW admits 11 million cars worldwide fitted with cheating devices. The CEO says he is "endlessly sorry" but will not resign. The US chief, Michael Horn, says the company "totally screwed up".

23 Sep The CEO quits but insists he is "not aware of any wrongdoing on his part". Class-action lawsuits are filed in the US and Canada and criminal investigations are launched by the US Justice Department.

24 Sep Officical confirms that VW vehicles with cheating software were sold across Europe as well. The UK Department for Transport says it will start its own inquiry into car emissions, as VW faces a barrage of legal claims from British car owners.

26 Sep Switzerland bans sales of VW diesel cars.

28 Sep German prosecutors launch an investigation of VW ex-CEO Winterkorn.

30 Sep Almost 1.2 million VW diesel vehicles in the UK are affected by the scandal, more than one in ten diesel cars on Britain's roads.

We have extended the Twitter dataset with a set of key phrases of length 2 that are potentially relevant to sentiment analysis. In this, we followed the approach discussed by Turney (2002). The main idea is to identify syntactic patterns that are considered suitable to matching subjective opinions (as opposed to objective facts). The resulting candidates for such *polarity_keywords* are linked in the database to the tweet from which they were extracted. This approach can be compared to another related approach to opinion extraction from financial news (Ruiz et al., 2012), in which sentiment gazetteers were also used to indicate the news polarity. Here the decision about polarity has not been made, but is left to future users of the data.

To extract the keywords in question, we employed the Stanford Part-Of-Speech (POS) tagger and Tgrep2 tool to extract the tag patterns proposed by Turney (2002), as listed in Table 2. About a third of all messages were annotated with pairs of key words as a result of the above mentioned procedure. In Table 3 we list the 20 most common pairs: on the whole, they appear quite specific and well correlated with the corpus topic.

In addition to the Twitter data, our dataset includes price information on the per-minute basis for Volkswagen (symbol: VOW.DE) shares and those of Ford (symbol: F) as an example of an automaker unaffected by the scandal. In addition, we have included S&P500 data (American Stock Market Index, symbol: SPY) as an indication of the state of the markets as a whole during the period in question. The data, as available from a number of public websites, includes time stamps, the 'open' and 'close' price, as well as the 'high' and 'low' price for the given one minute interval.

Figure 1 shows a comparison of Buy-and-hold[6] cumulative returns of those three securities during 15-30 Sep. 2015.

6. Ontology Representation and Sample Queries

The hierarchy of classes representing the dataset is shown in Figures 3. The **Event** class has three properties: *datetime*, *epoch* and *duration*. The *epoch* property is the number of seconds elapsed from 1st January 1970 00:00 UTC, which provides a common timeline between individuals. The *duration* property describes how long an event lasts and in our dataset, we use second as the timing unit. The **Event** class has two subclasses: **Tweet** and **OHLC**[7]. **Tweet** contains all the individuals storing tweets with their properties: *id*, *username*, *url*, *sourceUrl*, *numberOfRetweet* and *polarityKeyword*. **OHLC** contains all the individuals of stock price of specific company or market index. Each of them has the following properties: *high*, *low*, *open*, *close*, *symbol* and *isin*[8] (See Listing 1).

Listing 1: Individuals of **OHLC** and **Tweet**, shown in turtle format.

```
@prefix nsp: <http://example.org/vwevent2015/property/> .
@prefix nss: <http://example.org/vwevent2015/ontology/OHLC/> .
@prefix nst: <http://example.org/vwevent2015/ontology/Tweet/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

# An individual of OHLC
nss:f-1442323800 nsp:close "13.84"^^xsd:float ;
    nsp:datetime "2015-09-15T09:30:00-04:00" ;
    nsp:duration "60"^^xsd:xsd:unsignedLong ;
    nsp:epoch "1442323800"^^xsd:unsignedLong ;
    nsp:high "13.86"^^xsd:float ;
    nsp:low "13.79"^^xsd:float ;
    nsp:symbol "F" ;
    nsp:isin "US3453708600" ;
    nsp:open "13.8"^^xsd:float ;
    nsp:return "0.00289855072464"^^xsd:float .

# An individual of Tweet
<http://example.org/vwevent2015/ontology/Tweet/646575192907644928> nsp:
      datetime "2015-09-23T02:41:53-04:00" ;
    nsp:epoch "1442990513"^^xsd:unsignedLong ;
    nsp:id 646575192907644928 ;
    nsp:numberOfRetweet "0"^^xsd:unsignedLong ;
    nsp:polarityKeyword "criminal charges" ;
    nsp:sourceUrl <http://twitter.com/brian_poncelet/status
          /646575192907644928> ;
    nsp:url <http://twitter.com/Brian_Poncelet/status/646575192907644928> ;
    nsp:username "brian_poncelet" .
```

Representing our data as an ontology makes it possible to be queried in a flexible and powerful fashion, allowing its users to link the textual and time series data in a seamless way. Here are some examples of SPARQL queries seeking to extract useful features through the use of both polarity keywords and stock price movements.

Query 1 This SPARQL query will extract the tweets whose time stamp coincides with a drop in the Volkswagen stock price by more than 1%, ranked by *numberOfRetweets*.

The results of this query 1 are shown in listing 3. In order to improve readability, returns only show three decimal places, and *datetimes* are reformatted not to show the year.

[6] Buy-and-hold is a trading strategy, typically for benchmarking purposes, that considers the performance of buying the security and holding it for the whole period of analysis. Cumulative return on day i: $r_i = (price_i - price_{buy})/price_{buy}$.

[7] OHLC stands for open, high, low and close price of stock price during a period of time.

[8] ISIN refers to International Securities Identification Numbers, which provides a unique identification for each security.

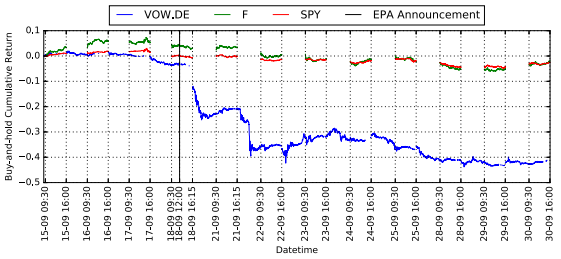


Figure 1: Buy-and-hold cumulative returns of Volkswagen stock, Ford stock and S&P500 during 15-30 September 2015.

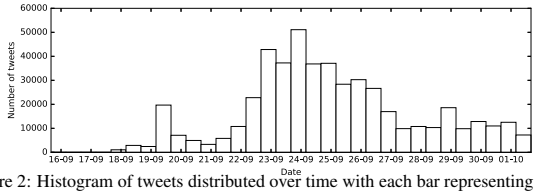


Figure 2: Histogram of tweets distributed over time with each bar representing 12 hours.

Listing 2: Query 1

```
PREFIX nsp: <http://example.org/vwevent2015/property/>
PREFIX nst: <http://example.org/vwevent2015/ontology/Tweet>
PREFIX nss: <http://example.org/vwevent2015/ontology/OHLC>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT
  ?username
  (?id AS ?tweet_id)
  ?return
  (?numberOfRetweet AS ?nbRt)
  ?datetime
  (group_concat(distinct ?pk;separator=", ") as ?polarityKeywords)
WHERE{
  ?ohlc nsp:epoch ?ohlc_epoch .
  ?ohlc nsp:return ?return .
  ?ohlc nsp:symbol "VOW.DE" .
  FILTER(?return < -0.01)
  ?tweet nsp:epoch ?tweet_epoch .
  ?tweet nsp:datetime ?datetime .
  ?tweet nsp:numberOfRetweet ?numberOfRetweet.
  ?tweet nsp:url ?url .
  ?tweet nsp:sourceUrl ?sourceUrl .
  ?tweet nsp:username ?username .
  ?tweet nsp:id ?id .
  ?tweet nsp:polarityKeyword ?pk .
  FILTER EXISTS{?tweet nsp:polarityKeyword ?pk}
  FILTER(
    ?url = ?sourceUrl
    && xsd:integer(?numberOfRetweet) >= 5
    && xsd:integer(?tweet_epoch) <= xsd:integer(?ohlc_epoch) + 60
    && xsd:integer(?tweet_epoch) >= xsd:integer(?ohlc_epoch)
  )
}
GROUP BY ?username ?id ?return ?numberOfRetweet ?datetime
ORDER BY DESC(xsd:integer(?numberOfRetweet)) ?return
LIMIT 10
```

Listing 3: Result of Query 1

	username	tweet_id	return	nbRt	datetime	polarityKeywords
1	business	646586797636644864	-0.023	113	09-23 03:28:00	as much
2	newswaala	646580334616645633	-0.011	30	09-23 03:02:19	high emissions first detected
3	twistools_en	646586860173688832	-0.023	8	09-23 03:28:15	national embarrassment
4	nytimesbusiness	646260916129005568	-0.022	6	09-22 05:53:04	diesel cars little effect
5	speedmonkeycouk	648435351476957184	-0.011	6	09-28 05:53:29	now being

Listing 4: Query 2

```
PREFIX nsp: <http://example.org/vwevent2015/property/>
PREFIX nst: <http://example.org/vwevent2015/ontology/Tweet>
PREFIX nss: <http://example.org/vwevent2015/ontology/OHLC>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT
  ?pk
  (COUNT(?pk) AS ?count)
WHERE{
  {
    SELECT
      (xsd:unsignedLong(xsd:float(?ohlc_epoch)/60.0) AS ?ohlc_minute)
      (xsd:unsignedLong(xsd:float(?tweet_epoch)/60.0+1.0) AS ?tweet_minute)
      ?pk
      ?return
    WHERE{
      ?ohlc nsp:epoch ?ohlc_epoch ;
        nsp:return ?return ;
        nsp:symbol "VOW.DE" .
      FILTER(?return <= -0.02)
      ?tweet nsp:epoch ?tweet_epoch ;
        nsp:polarityKeyword ?pk ;
    }
    HAVING(?ohlc_minute=?tweet_minute)
  }
}
GROUP BY ?pk
ORDER BY DESC(?count)
LIMIT 20
```

Listing 5: Result of Query 2

	pk	count
1	worldwide fitted	41
2	as much	23
3	entire auto	14
4	sure people	11
5	first detected	10
6	high emissions	10
7	multiple probes	7
8	totally screwed	7
9	chief executive	5
10	diesel cars	5
11	early trading	5
12	here come	5
13	national embarrassment	5
14	little effect	4
15	not sure	4
16	also installed	3
17	false emission	3
18	internal investigations	3
19	just lost	3
20	absolutely foolish	2

Expression	Word1	Word2	followed by
(JJ.(NN|NNS))	JJ	NN or NS	no restrictions
(RB.(JJ!.(NN|NNS)))	RB	JJ	not NN nor NNS
(RBR.(JJ!.(NN|NNS)))	RBR	JJ	not NN nor NNS
(RBS.(JJ!.(NN|NNS)))	RBS	JJ	not NN nor NNS
(JJ.(JJ!.(NN|NNS)))	JJ	JJ	not NN nor NNS
(NN.(JJ!.(NN|NNS)))	NN	JJ	not NN nor NNS
(NS.(JJ!.(NN|NNS)))	NS	JJ	not NN nor NNS
(RB.(VB|VBD|VBN|VBG))	RB	VB, VBD, VBN or VBG	no restrictions
(RBR.(VB|VBD|VBN|VBG))	RBR	VB, VBD, VBN or VBG	no restrictions
(RBS.(VB|VBD|VBN|VBG))	RBS	VB, VBD, VBN or VBG	no restrictions

Table 2: Extracted Word1+Word2 keyphrases using *Tgrep2* expressions

keywords	count		
diesel scandal	3993	diesel deception	1294
chief executive	3835	multiple probes	1189
diesel emissions	3280	electric car	1166
diesel cars	2980	new tech	1110
sure people	2801	clean diesel	1059
new boss	2407	criminal probe	1037
totally screwed	2208	finally be	953
clean air	1919	fresh start	908
as many	1449	refit cars	898
criminal charges	1323	diesel vehicles	890

Table 3: 20 most common pairs of keywords extracted from the Twitter data.

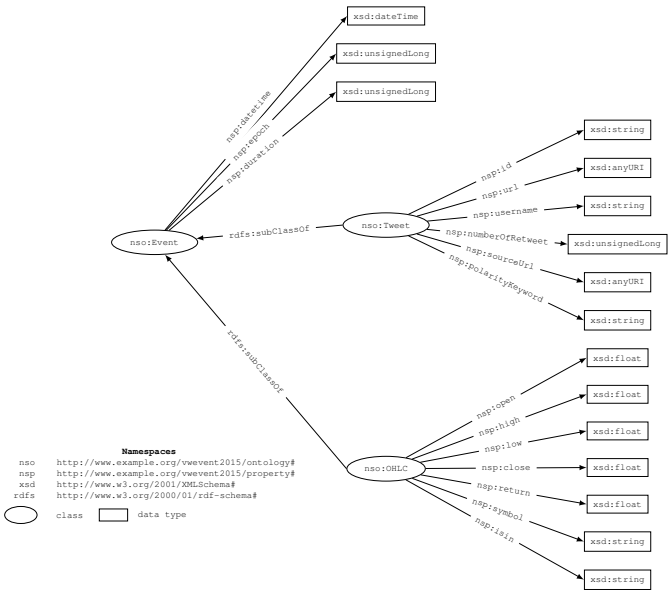


Figure 3: VW Event Ontology Classes

The first tweet was published by Bloomberg (@business):

> CEO Martin Winterkorn faces a showdown with #Volkswagen's board later http://bloom.bg/1FdA4sA

Tweet No. 4 came from Business news of NY Times (@nytimesbusiness):

> Volkswagen's recall troubles may have little effect on China: It sells almost no diesel cars in the country. http://nyti.ms/1Jmipd8

Apart from main public media accounts, we found that among the authors of those tweets are also an indian media (No. 2), a marketing account (No. 3), a motor amateur (No. 5). This indicates our dataset contains information from a range of sources that provide potentially useful information on this event.

Query 2 We have also been able to check whether some of the keywords are associated with specific stock price movements by using the following SPARQL query, which aims to retrieve the keywords associated on drops in Volkswagen price greater than 2% within any one-minute-period.

The result of Query 2 shows that in most cases, the worst drops in VW price coincide with keywords expressing negative sentiment or referring to some of the specific facts of the scandal (e.g. "worldwide fitted", "diesel cars").

Query 3 For users with access to twitter contents (mapped to nsp:content), listing 6 shows the potential usage of connecting with other existing ontologies to combine domain knowledge with stock price time series: *get the average one minute return of stock the surname of a key person* (CEO for example) *appears in the tweets.*

Listing 6: Query 3

```
PREFIX nsp: <http://example.org/vwevent2015/property/>
PREFIX nst: <http://example.org/vwevent2015/ontology/Tweet>
PREFIX nss: <http://example.org/vwevent2015/ontology/OHLC>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX db: <http://dbpedia.org/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT
  ?sn (AVG(?return) AS ?avgReturn)
WHERE{
  SERVICE <http://dbpedia.org/sparql/>{
    ?company dbo:keyPerson ?person .
    ?person foaf:surname ?surname .
    BIND(LCASE(STR(?surname)) AS ?sn)
    FILTER(?company=<http://dbpedia.org/resource/Volkswagen>)
  }
  ?ohlc nsp:epoch ?ohlc_epoch .
  ?ohlc nsp:return ?return .
  ?ohlc nsp:symbol "VOW.DE" .
  ?tweet nsp:epoch ?tweet_epoch .
  ?tweet nsp:content ?content .
  ?tweet nsp:url ?url .
  ?tweet nsp:sourceUrl ?sourceUrl .
  ?tweet nsp:id ?id .
  ?tweet nsp:numberOfRetweet ?numberOfRetweet
  FILTER(?url=?sourceUrl && ?numberOfRetweet > 100)
  FILTER(CONTAINS(LCASE(?content), ?sn))
  FILTER(
    xsd:integer(?ohlc_epoch) >= xsd:integer(?tweet_epoch) &&
    xsd:integer(?ohlc_epoch) <= xsd:integer(?tweet_epoch) + 60
  )
}
GROUP BY ?sn
```

7. Conclusion and Future Works

With the advantages of ontology representation, discovering useful information in time-labelled text data (tweets) and numerical time series (stock prices) becomes an easier task. Both queries and dataset can be easily modified or extended. On the other hand, copyright issues with Twitter data put limits to displaying and sharing information in a more straightforward way, and restrict us to only displaying tweet IDs in our dataset.

The polarity keywords are a useful feature, despite the unsupervised way in which they were extracted. Our future work will focus on adding to the range of features available in the dataset.

We also want to assess our work in connection with other related ontologies for stock markets[9] (Alonso et al., 2005) and companies[10] as described in DBpedia. Such integration for example should allow one to recognise Volkswagen Group as an entity of Public Company in DBpedia[11], where we can find information about their assets, revenue, owner, holding company, products and many more. This type of information would potentially allow one to automatically link one company affected by adverse events to, say, its subsidiary companies, which one may expect also to feel the repercussions of such events. Indeed, Audi, Seat and Škoda, all subsidiary companies of VW Group, were all eventually linked to the diesel engine cheating software scandal. More recent news from France has shown that any results from our data could also find use to handle other related news from the automotive industry. We hope that our work will encourage more interesting research in the financial domain as a whole.

8. References

Alonso, L., Bas, L., Bellido, S., Contreras, J., Benjamins, R., and Gomez, M. (2005). WP10: Case Study eBanking D10. 7 Financial Ontology. *Data, Information and Process Integration with Semantic Web Services, FP6-507483*.

Asur, S. and Huberman, B. A. (2010). Predicting the Future with Social Media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference*, volume 1, pages 492–499. IEEE.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter Mood Predicts the Stock Market. *Journal of Computational Science*, 2(1):1 – 8.

Butler, M. and Kazakov, D. (2012). Testing Implications of the Adaptive Market Hypothesis via Computational Intelligence. In *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2012 IEEE Conference on*, pages 1–8. IEEE.

Cotfas, L.-A., Delcea, C., Roxin, I., and Paun, R., (2015). *New Trends in Intelligent Information and Database Systems*, chapter Twitter Ontology-Driven Sentiment Analysis, pages 131–139. Springer International Publishing, Cham.

[9] http://dbpedia.org/page/Stock_market
[10] http://dbpedia.org/ontology/company
[11] http://dbpedia.org/resource/Volkswagen

Eichstaedt, J. C., Schwartz, H. A., Kern, M. L., Park, G., Labarthe, D. R., Merchant, R. M., Jha, S., Agrawal, M., Dziurzynski, L. A., Sap, M., et al. (2015). Psychological Language on Twitter Predicts County-level Heart Disease Mortality. *Psychological Science*, 26(2):159–169.

Fama, E. F. (1965). The Behavior of Stock-market Prices. *Journal of Business*, 38(1):34–105.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *CS224N Project Report, Stanford*, 1:12.

Grossman, S. J. and Stiglitz, J. E. (1980). On the Impossibility of Informationally Efficient Markets. *The American Economic Review*, pages 393–408.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220.

Haugen, R. A. (1995). *The New Finance: the Case Against efficient markets*. Prentice Hall Englewood Cliffs, NJ.

Kollewe, J. (2015). Volkswagen Emissions Scandal Timeline. http://www.theguardian.com/business/2015/dec/10/volkswagen-emissions-scandal-timeline-events Accessed: Jan. 08, 2016.

Kontopoulos, E., Berberidis, C., Dergiades, T., and Bassiliades, N. (2013). Ontology-based Sentiment Analysis of Twitter Posts. *Expert Systems with Applications*, 40(10):4065–4074.

Lee, C.-H. and Wu, C.-H. (2015). Extracting Entities of Emergent Events from Social Streams Based on a Data-Cluster Slicing Approach for Ontology Engineering. *International Journal of Information Retrieval Research*, 5(3):1–18, July.

Lupiani-Ruiz, E., García-Manotas, I., Valencia-García, R., García-Sánchez, F., Castellanos-Nieves, D., Fernández-Breis, J. T., and Camón-Herrero, J. B. (2011). Financial News Semantic Search Engine. *Expert Systems with Applications*, 38(12):15565–15572.

Mellouli, S., Bouslama, F., and Akande, A. (2010). An Ontology for Representing Financial Headline News. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(2–3):203–208.

O'Connor, B., Balasubramanyan, R., Routledge, B. R., and Smith, N. A. (2010). From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. *ICWSM*, 11(122-129):1–2.

Piskorski, J. and Yangarber, R. (2013). Information Extraction: Past, Present and Future. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 23–49. Springer.

Ruiz, E. J., Hristidis, V., Castillo, C., Gionis, A., and Jaimes, A. (2012). Correlating Financial Time Series with Micro-blogging Activity. *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining - WSDM '12*, page 513.

Shiller, R. J. (2003). From Efficient Markets Theory to Behavioral Finance. *Journal of Economic Perspectives*, pages 83–104.

Shleifer, A. (2000). *Inefficient Markets: An Introduction to Behavioral Finance*. Oxford University Press.

Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., and Deng, X. (2013). Exploiting Topic based Twitter Sentiment for Stock Prediction. In *ACL (2)*, pages 24–29.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. (2010). Predicting Elections with Twitter: What 140 Characters Reveal About Political Sentiment. *ICWSM*, 10:178–185.

Turney, P. D. (2002). Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wolfram, M. S. A. (2010). *Modelling the Stock Market using Twitter*. Master thesis, The University of Edinburgh.

Yang, J. and Leskovec, J. (2011). Patterns of Temporal Variation in Online Media. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pages 177–186. ACM.

Zhang, X., Fuehres, H., and Gloor, P. A. (2011). Predicting Stock Market Indicators Through Twitter "I hope it is not as bad as I fear". *Procedia-Social and Behavioral Sciences*, 26:55–62.

Appendix B

Quantifying Correlation between Financial News and Stocks

Quantifying Correlation between Financial News and Stocks

Haizhou Qu
Department of Computer Science, University of York
York, United Kingdom, YO105GH
hq524@york.ac.uk

Dimitar Kazakov
Department of Computer Science, University of York
York, United Kingdom, YO105GH
dimitar.kazakov@york.ac.uk

***Abstract*—Financial news and stocks appear linked to the point where the use of online news to forecast the markets has become a major selling point for some traders. The correlation between news content and stock returns is clearly of interest, but has been mostly centred on news meta-data, such as volume and popularity. We address this question here by measuring the correlation between the returns of 27 publicly traded companies and news about them as collected from Yahoo Financial News for the period 1 Oct 2014 to 30 Apr 2015. In all reported experiments, two metrics are defined, one to measure the distance between two time series, the other to quantify the difference between two collections of news items. Two 27 $\times$ 27 distance matrices are thus produced, and their correlation measured with the Mantel test. This allows us to estimate the correlation of stock market data (returns, change, volume and close price) with the content of published news in a given period of time. A number of representations for the news are tested, as well as different distance metrics between time series. Clear, statistically significant, moderate level correlations are detected in most cases. Lastly, the impact of the length of the period studied on the observed correlation is also investigated.**

I. Introduction

Using online financial news to assist forecasts using time series is very tempting as one's intuition suggests that there should be useful information in the news that is not directly reflected in the day-to-day figures reported for each publicly traded company. Indeed, there have been stock market traders, such as the now defunct Derwent Capital Markets hedge fund [1], which advertise that their forecasting algorithms make use of online social media, such as Twitter [2]. Nevertheless, any attempt to make use of such information brings up some difficult questions: What is the most useful representation of the text documents to be used? What features should one extract from them, and in what form? Can we decide when news is useful, if at all? Indeed, in addition to the implications of economic theories, such as the efficient market hypothesis, which seem to imply that the time series data available to traders only contains noise, one can also consider the case in which news only follows the markets with a certain delay, which would render it useless. These are hard questions and the road to answering them would be made easier if split into several stages.

We have previously looked at one particular financial event, the price crash of Volkswagen stock that followed the announcement of the US Environment Protection Agency investigation into what became known as the Dieselgate scandal. This was done in the hope that such an extreme event (where a substantial and sustained decline in price follows a news release) could provide an excellent data set on which to study the likely impact of financial news about a company on its performance on the stock market [3], with a focus on the potential causal link.

Here we change the perspective and instead want to study whether for a given time period, news and stock market data are correlated, leaving out the chicken-and-egg question of which one came first. In addition, we also ignore the exact time of news release, and combine all news about a given company published within the time period of interest into a single document. Then we study the differences between the news about a pair of companies, and how well such a difference is (co-)related to a difference in the performance of the two stocks over the same period. The chosen statistical measure, namely, the Mantel test, measures the correlation between differences in the news and in the time series for a whole set of companies at once, which should make the results less dependant on the circumstances of each individual company.

II. Data

We collected online news from Yahoo Financial News over the period 1 Oct 2014 – 30 Apr 2015. Each news item carries a time stamp (in EST time) and the symbols of one or more companies, to which it is related. The 27 stocks were selected to have no more than a total of 5 days with no news about them in the studied period. Very short news with less than 10 words or 100 characters were ignored, leaving a total of 67,840 news items.

We have also collected daily stock market data for the same companies and period of time. For each day and stock, the data set contains the $open$, $high$, low and $close$ price, as well as the $volume$ traded and the $adjusted\ close$ price. Table III shows a summary of the data available.

III. Method

A. The Mantel Test

The Mantel test is a statistical test to determine correlation between two pairwise distance matrices with the same rank [4].

Given two $n \times n$ distance matrices $\mathbf{U}$ and $\mathbf{V}$, it calculates the correlation r using equation 1.

$$r_{\mathbf{U},\mathbf{V}} = \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{U_{ij} - \bar{u}}{\sigma_{\mathbf{U}}} \cdot \frac{V_{ij} - \bar{v}}{\sigma_{\mathbf{V}}} \quad (1)$$

where $m = n(n-1)/2$ is the number of pairwise distances of a size n population, $\bar{u}$ and $\bar{v}$ are means of pairwise distance elements located in upper triangle exclude the diagonal of $\mathbf{U}$ and $\mathbf{V}$.

For example, the matrix $\mathbf{U}$ may represent the genetic distances in a group of n species while another matrix $\mathbf{V}$ represents the geographic distance between the species' habitats. By applying the Mantel test, we can calculate how much the geographic distance between two species is correlated with their genetic differences. The idea behind the Mantel test is to randomly permute one matrix repeatedly while calculating each time a correlation according to equation 1. A hypothesis test is carried out to examine if this correlation is significantly lower than the one produced with the original matrices. Finding that this is the case would suggest a correlation between the two matrices, and the two sets of distances they represent. As already mentioned, we use the Mantel test here to determine the extent to which financial news and stocks are correlated with each other. In other words, we wanted to see whether the differences between a pair of time series, e.g. representing the daily returns of two stocks, are correlated with the differences in the news about these companies. For this purpose, it was necessary to define ways to measure the difference (or *distance*) between time series, and between text documents.

B. Comparing Time Series

We have considered three distance metrics to compare pairs of time series: the Cosine Distance (CD), the Euclidean Distance (ED) and the one produced by Pearson's correlation (further referred to as PD).

The well known cosine distance is calculated according to equation 2 where u_t and v_t are the values of each time series on day t.

Euclidean distance is determined by the length of the line segment connecting points $\mathbf{u}$ and $\mathbf{v}$ (see equation 3). It is a proper distance metric with wide-ranging spectrum of applications.

Pearson's correlation measures the linear correlation between a pair of variables (time series in our case) through the ratio of their covariance divided by the product of their standard deviations. Equation 4 shows how a distance metric can be defined on the basis of this correlation (see equation 4).

$$CD(\mathbf{u}, \mathbf{v}) = 1 - \frac{\sum_t v_t \cdot u_t}{\sqrt{\sum_t u_t^2 \cdot \sum_t v_t^2}} \quad (2)$$

$$ED(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_t (u_t - v_t)^2} \quad (3)$$

$$PD(\mathbf{u}, \mathbf{v}) = 1 - \frac{\sum_t (u_t - \bar{u})(v_t - \bar{v})}{\sqrt{\sum_t (u_t - \bar{u})^2 \sum_t (v_t - \bar{v})^2}} \quad (4)$$

where $\mathbf{u}$ and $\mathbf{v}$ are two vectors representing two time series with the same time index.

C. Comparing Texts

There is a number of representations developed for the purposes of Information Retrieval that could be used in this study. These range from the simplest bag-of-words model, which only takes into account the presence (and frequency) of words in a document, but ignores any word order, to representations of words and their neighbours (bigrams, trigrams, etc.) and those in which parts of the parse tree of a sentence are used as features [5].

A Bag-of-Words represents a collection of texts as a `document × word` matrix which treats each word in the whole collection as a separate feature. The content of each document is then encoded as a vector containing the (relative) frequency of each of its words, including zeros for all the words that do not appear in the document. This allows for an easy comparison between any two documents, at the price of ignoring the grammatical relationship between words. So, a set of text documents $\mathcal{D}$ is represented as a matrix M where each row corresponds a document $d \in \mathcal{D}$, and each column stands for a feature w (usually a word or token). Each element $M_{i,j}$ then is the relative frequency with which word j appears in document i.

An additional weighting scheme is often used to reduce the importance of words that appear across most documents, and highlight the ones that are characteristic to a small subset of documents. TF-IDF (Term Frequency – Inverse Document Frequency) [6] is the most popular such technique. Here the relative frequency of word w in document d is weighted according to equation 5. This reduces the perceived importance of a word w in a document d to zero if the word appears in all documents, and increases it gradually as the number of documents containing w decreases [7].

$$tfidf_{d,w} = \frac{freq_{w,d}}{|d|} \cdot log \frac{|\mathcal{D}|}{|\{d \in \mathcal{D} : w \in d\}|} \quad (5)$$

where $|\cdot|$ is size of a set; $freq_{w,d}$ is the number of occurrences of word w in document d.

As the number of words in a large document collection could surpass 10^6 (which would result in up to 10^{12} possible bigrams, if these were used), dimension-reducing techniques can also be considered to reduce the dimensionality of the representation in order to fight increase in computational complexity and sparsity of data. One such approach that is quickly growing in popularity is *word2vec* [8], which uses the class of neural networks popularised under the label of Deep Learning to reduce the representation dimensionality to value k which is typically $100 < k < 1000$. The result is that each word is represented as a linear combination of these new features, that is, a vector of size k known as *word embedding*. We then represent a document of n words as the average of its n word embeddings. A set of m documents is then represented as a matrix of size $m \times k$. The method relies on distributional statistics of words within a fixed-size window. These are often

collected from very large corpora and then used with other documents of interest.

In this study, we always preprocess all text documents in the following way. First, the text is *tokenized*, i.e. split into separate words or punctuation symbols. Then we remove all punctuation and *stop words*, essentially all pronouns, prepositions, conjunctions and a few very common verbs. The remaining words are *lemmatized*, i.e. replaced by their standard entry in the dictionary. All URLs are then mapped to the same string (URL), email addresses are mapped to the string EMAIL, and numbers are mapped to NUM. Finally, we merge all preprocessed news items for each company into a single document. From this, we produce two representations of all news on m companies making use of n different words. One is the TF-IDF weighted bag-of-words $m \times n$ matrix, the other – the $m \times k$ matrix produced with the word2vec approach (where $k = 300$). For each of these representations we have experimented with two different distance metrics, CD and ED (as defined in Section III-B) to produce four different distance matrices representing how the news about our 27 companies differ from each other.

D. Modelling Stock Prices

Our data set contains the daily $open$, $high$, low, $close$ prices for each company, as well and the $volume$ of trade on that day.

Here $close$ refers to the final price of last deal before the stock market closes and we are using $adjusted\ close$, which refers to the price that depicts the effects of corporation actions such as dividends and stock split. $high$, resp. low refers to the highest, resp. lowest price achieved during the day. $volume$ is total number of shares traded on that day. We have also calculated the $overnight\ return$ according to equation 6 and $change$ according to equation 7.

$$return_t = (close_t - close_{t-1})/close_{t-1} \quad (6)$$

$$change_t = (high_t - low_t)/open_t \quad (7)$$

In this study we have used in turn data on $close$, $volume$, $return$ and $change$ to produce distance matrices in each case using each of the three distance metrics defined in Section III-B. In each sliding window, the $close$ and $volume$ time series were *standarlized* according to the equation $s' = (s_t - \bar{s})/\sigma_{\mathbf{s}}$. Given a window from t_{start} to t_{end}, each variable will be represented as a vector: $< s'_{t_{start}}, s'_{t_{start}+1}, \cdots, s'_{t_{end}} >$

TABLE I: Experimental Settings

Setting	Options
News Representation	$tfidf$, $word2vec$
Text Distance Metric	CD, ED
Time Series	$close$, $volume$, $return$, $change$
Time Series Distance Metric	CD, ED, PD

$\mathbf{u}_x$: bag-of-words representation of news about stock x (e.g. $AAPL$)
$\mathbf{u}_y$: bag-of-words representation of news about stock y (e.g. $GOOG$)
$\mathbf{v}_x$: time series of stock x
$\mathbf{v}_y$: time series of stock y
$\mathbf{U}$: distance matrix of news
$\mathbf{V}$: distance matrix of stock time series

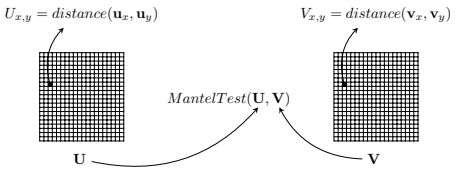


Fig. 1: Illustration of one Mantel test for an observation period.

IV. Experiment Design and Results

The first set of experiments considers all data available from 1 Oct 2014 until 30 April 2015. For each combination of time series, time series distance metric, text representation and text distance metric (see Table I), we performed the Mantel test to measure the correlation between news and time series. The results for all 48 combinations of experimental settings are reported in Table II along with the p-value of each test.

In our final set of experiments, we wanted to see whether the length of the time period used in the tests affected the levels of correlation, and to what extent the correlation varied over time. For that purpose, we used a 28-day long sliding window, and gradually shifted it with a 1-day step to produce 185 samples. All 48 experimental settings were then applied in turn to each sample, with the results plotted in the form of graphs, as shown in Figures 2 and 3.

V. Discussion

The results with the full data set suggest low to moderate levels of statistically significant correlation between news and financial performance, which for the best set of parameters reaches values of around 0.44. This, of course, does not indicate whether it is the news that affects the prices or, for instance, whether the news does not simply reflect the numerical data with a certain delay, which is likely to render it useless for forecasting. We do not attempt to answer this question here. On the other hand, the levels of correlation, achieved without any optimisation process in the choice of text-based features should serve as an encouragement to further studies, in which the most useful text features, be it words, bigrams, syntactic trees, etc. could be detected, either for the whole area of financial forecasting, or for a selected set of sectors.

The results show that text representations using $tfidf$ consistently outperform $word2vec$ and result in higher Mantel correlations between text and time series. $volume$ (*standarlized*) shows the highest correlation with news when $tfidf$ is used. Overnight $return$ also shows significant correlation with news of 0.35.

The results with the 28-day sliding window data show statistically significant results for extended periods of time with the correlation reaching levels of over 0.45 in some cases. There is a difference among the 4 variables representing stocks, with the *standardlized* $volume$ again showing the strongest correlation over the longest periods of time. Overnight returns also show substantial levels of correlation with news, albeit less often, and to a lower degree. It is also very interesting to observe sharp changes in the correlation levels, which could be potentially useful to detect important events as the first step towards forecasts that take into account the effect of such externalities.

It is worth mentioning that according to Augmented Dickey Fuller test, the $close$ price of all 27 stocks in our long observation time period is not stationary or trend stationary, thus PD is not a suitable distance metric, since the presence of trends in a pair of time series will boost the levels of correlation reported. We have kept these figures here for the sake of completeness.

$tfidf$ outperforms $word2vec$ with higher Mantel correlations, and, in the case of the sliding window data, also yields longer periods of statistical significance. Nevertheless, the plots also show that there are times when $word2vec$ is better than $tfidf$ at capturing significant correlations.

TABLE II: Correlation between news and financial time series as measured by the Mantel test according to subsection III-A ($10,000$ permutations for each Mantel test). A p-value for each Mantel test is also reported (in brackets); p-values less than 0.05 are shown in bold.

		$tfidf$		$word2vec$	
		CD	ED	CD	ED
$close$	CD	0.0820 (0.3947)	0.1581 (0.1210)	0.0108 (0.9006)	0.0196 (0.8061)
	ED	0.0922 (0.3413)	0.1578 (0.1364)	0.0253 (0.7797)	0.0373 (0.6558)
	PD^*	0.0820 (0.3905)	0.1581 (0.1257)	0.0108 (0.9028)	0.0196 (0.8182)
$volume$	CD	**0.4214 (0.0003)**	**0.4054 (0.0012)**	0.0084 (0.9374)	0.0591 (0.5765)
	ED	**0.4433 (0.0001)**	**0.4232 (0.0006)**	0.0230 (0.8225)	0.0799 (0.4231)
	PD	**0.4214 (0.0002)**	**0.4054 (0.0016)**	0.0084 (0.9347)	0.0591 (0.5595)
$return$	CD	**0.3553 (0.0013)**	**0.3476 (0.0043)**	0.1695 (0.0855)	**0.1998 (0.0388)**
	ED	0.2105 (0.1983)	0.2507 (0.1632)	-0.0459 (0.7331)	-0.0338 (0.7960)
	PD	**0.3567 (0.0007)**	**0.3479 (0.0051)**	0.1687 (0.0947)	**0.1991 (0.0393)**
$change$	CD	0.1724 (0.2465)	0.1673 (0.3110)	0.2415 (0.0650)	0.1122 (0.3587)
	ED	0.2839 (0.0663)	**0.3555 (0.0338)**	0.1190 (0.3654)	0.0149 (0.9093)
	PD	**0.3491 (0.0036)**	**0.3341 (0.0121)**	0.0484 (0.6669)	0.0961 (0.3618)

(* PD is not stationary)

REFERENCES

[1] D. Tweney, "Twitter-fueled hedge fund bit the dust, but it actually worked," http://venturebeat.com/2012/05/28/twitter-fueled-hedge-fund-bit-the-dust-but-it-actually-worked/, 2012, [Online; accessed 12-Oct-2016].

[2] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.

[3] H. Qu, M. Sardelich, N. N. Qomariyah, and D. Kazakov, "Integrating time series with social media data in an ontology for the modelling of extreme financial events," in *LREC 2016 Joint Second Workshop on Language and Ontology & Terminology and Knowledge Structures*, 2016.

[4] N. Mantel, "The detection of disease clustering and a generalized regression approach," *Cancer research*, vol. 27, no. 2.

[5] A. Moschitti, "Efficient convolution kernels for dependency and constituent syntactic trees," in *European Conference on Machine Learning*. Springer, 2006, pp. 318–329.

[6] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[7] J. Sedding and D. Kazakov, "Wordnet-based text document clustering," in *COLING 2004 3rd Workshop on Robust Methods in Analysis of Natural Language Data*, V. Pallotta and A. Todirascu, Eds., Geneva, 2004.

[8] T. Mikolov and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, 2013.

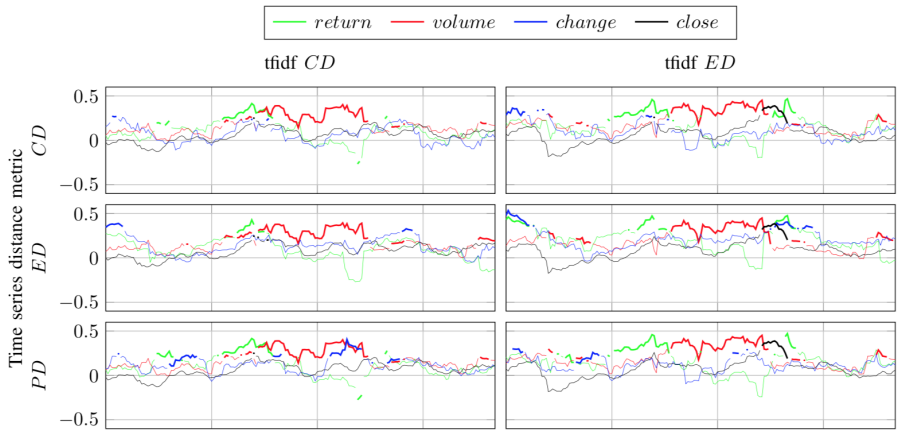


Fig. 2: Mantel test correlations for a sliding window of 4 weeks. Bold lines indicate statistically significant results ($p < 0.05$).

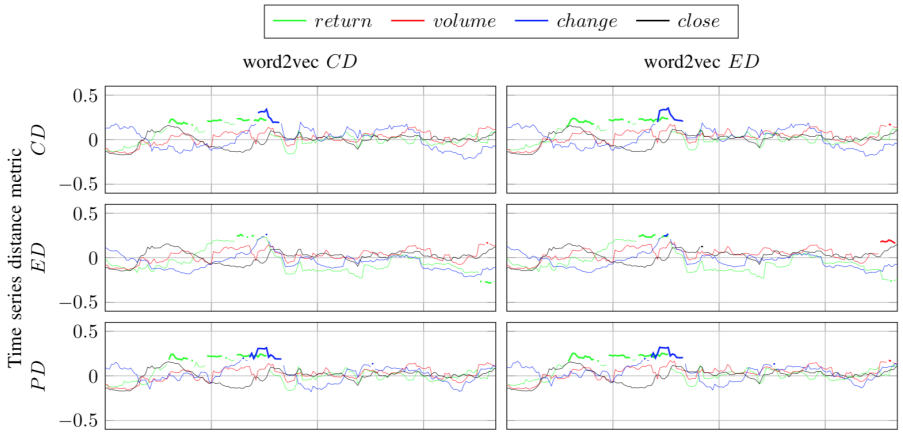


Fig. 3: Mantel test correlations for a sliding window of 4 weeks. Bold lines indicate statistically significant results ($p < 0.05$).

TABLE III: Information for each company: symbol, name, stock exchange, close price and number of news per day.

Symbol	Name	Exchange	Stock Price	Number of News
AAPL	Apple Inc.	NASDAQ		
AMZN	Amazon.com Inc.	NASDAQ		
BA	Boeing Co.	NYSE		
CMCSA	Comcast Co.	NASDAQ		
CSCO	Cisco Systems, Inc.	NASDAQ		
CVX	Chevron Co.	NYSE		
DIS	Walt Disney Co.	NYSE		
EBAY	eBay Inc.	NASDAQ		
FB	Facebook Common Stock	NASDAQ		
GE	General Electric Co.	LON		
GOOG	Alphabet Inc. Class C	NASDAQ		
GOOGL	Alphabet Inc. Class A	NASDAQ		
GS	Goldman Sachs Group Inc.	NYSE		
HD	Home Depot Inc.	NYSE		
IBM	IBM Common Stock	LON		
INTC	Intel Co.	NASDAQ		
JPM	JPMorgan Chase & Co.	NYSE		
KO	The Coca-Cola Co.	NYSE		
MSFT	Microsoft Co.	NASDAQ		
NFLX	Netflix, Inc.	NASDAQ		
NKE	Nike Inc.	NYSE		
SBUX	Starbucks Co.	NASDAQ		
T	AT&T Inc.	NYSE		
TSLA	Tesla Motors Inc.	NASDAQ		
VZ	Verizon Communications Inc.	NYSE		
WMT	Wal-Mart Stores, Inc.	NYSE		
YHOO	Yahoo! Inc.	NASDAQ		

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Arias, M., Arratia, A., and Xuriguera, R. (2014). Forecasting with Twitter Data. *ACM Transactions Intelligence System Technology*, 5(1):8:1—-8:24.

Basu, S. (1977). Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis. *The journal of Finance*, 32(3):663–682.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155.

Bird, S. (2006). NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, COLING-ACL 06, pages 69–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Boudoukh, J., Feldman, R., Kogan, S., and Richardson, M. (2013). Which News Moves Stock Prices? A Textual Analysis. *NBER Working Paper Series*, pages 1–45.

Butler, M. and Kazakov, D. (2012). Testing Implications of the Adaptive Market Hypothesis via Computational Intelligence. In *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2012 IEEE Conference on*, pages 1–8. IEEE.

Buttigieg, P. L. and Ramette, A. (2014). A guide to statistical analysis in microbial ecology: a community-focused, living review of multivariate data analyses. *FEMS Microbiology Ecology*, 90(3):543–550.

Chan, K. C., Gup, B. E., and Pan, M.-S. (1997). International stock market efficiency and integration: A study of eighteen nations. *Journal of Business Finance & Accounting*, 24(6):803–813.

Chen, K., Zhou, Y., and Dai, F. (2015). A lstm-based method for stock returns prediction: A case study of china stock market. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2823–2824. IEEE.

Chong, E., Han, C., and Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, 83:187–205.

De Boom, C., Van Canneyt, S., Demeester, T., and Dhoedt, B. (2016). Representation learning for very short texts using weighted word embedding aggregation. pages 1–8.

De Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–4592.

Ding, X., Zhang, Y., Liu, T., and Duan, J. (2015). Deep learning for event-driven stock prediction. *IJCAI International Joint Conference on Artificial Intelligence*, 2015-Janua(Ijcai):2327–2333.

Dingare, S., Nissim, M., Finkel, J., Manning, C., and Grover, C. (2005). A system for identifying named entities in biomedical text: How results from two evaluations reflect on both the system and the evaluations. *Comparative and Functional Genomics*, 6(1-2):77–85.

Fama, E. F. (1970). Efficient Capital Markets: A Review Of Theory And Empirical Work. *The Journal of Finance*, 25(2):383–417.

Fiesler, E. (1994). Comparative bibliography of ontogenic neural networks. In Marinaro, M. and Morasso, P. G., editors, *ICANN '94*, pages 793–796, London. Springer London.

Foucault, T., Hombert, J., and Rosu, I. (2012). News Trading and Speed. *Finance Seminar*, pages 1–60.

Fung, G. P. C., Yu, J. X., and Lu, H. (2005). The predicting power of textual information on financial markets. *IEEE Intelligent Informatics Bulletin*, 5(1):1–10.

Gidofalvi, G. and Elkan, C. (2001). Using news articles to predict stock price movements.

Hayek, F. A. (1945). The use of knowledge in society. *The American economic review*, 35(4):519–530.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. *Arxiv.Org*, 7(3):171–180.

Honnibal, M. and Montani, I. (2017). spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Hu, Z., Liu, W., Bian, J., Liu, X., and Liu, T.-Y. (2017). Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction. pages 261–269.

Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Joachims, T. (1999). Making large scale SVM learning practical.

Junqué De Fortuny, E., De Smedt, T., Martens, D., and Daelemans, W. (2014). Evaluating and understanding text-based stock price prediction models. *Information Processing and Management*, 50(2):426–441.

Kenter, T., Borisov, A., and de Rijke, M. (2016). Siamese CBOW: Optimizing Word Embeddings for Sentence Representations. pages 941–951.

Keogh, E. and Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping Eamonn. *Knowledge and Information Systems*, 7:358–386.

Ko, S., Cho, I., Afzal, S., Yau, C., Chae, J., Malik, A., Beck, K., Jang, Y., Ribarsky, W., and Ebert, D. S. (2016). A Survey on Visual Analysis Approaches for Financial Data. *Computer Graphics Forum*, 35(3):599–617.

Kogan, S., Levin, D., Routledge, B. R., Sagi, J. S., and Smith, N. A. (2009). 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*, pages 272–280. Association for Computational Linguistics.

Komninos, A. and Manandhar, S. (2016). Dependency Based Embeddings for Sentence Classification Tasks. *Naacl2016*, pages 1490–1500.

Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. (2015). From Word Embeddings To Document Distances. *Proceedings of The 32nd International Conference on Machine Learning*, 37:957–966.

Lab, L. (2017). Convolutional Neural Networks (LeNet). http://deeplearning.net/tutorial/lenet.html. [Online; accessed 12-Jan-2017].

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., and Allan, J. (2000). Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining*, pages 37–44. Citeseer.

Le, A. N., Martinez, A., Yoshimoto, A., and Matsumoto, Y. (2017). Improving Sequence to Sequence Neural Machine Translation by Utilizing Syntactic Dependency Information. *IJCNLP17*, pages 21–29.

Lee, C. H. L., Liu, A., and Chen, W. S. (2006). Pattern discovery of fuzzy time series for financial prediction. *IEEE Transactions on Knowledge and Data Engineering*, 18(5):613–625.

Leigh, W., Frohlich, C. J., Hornik, S., Purvis, R. L., and Roberts, T. L. (2008). Trading with a stock chart heuristic. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 38(1):93–104.

Li, B. and Hoi, S. C. H. (2012). Online Portfolio Selection: A Survey. V(212).

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

Loughran, T. and McDonald, B. (2011). "when is a liability not a liability? textual analysis, dictionaries, and 10-ks". *The Journal of Finance*, 66(1):35–65.

Malhotra, P. (2017). ResNet50-Tensorflow. https://github.com/piyush2896/ResNet50-Tensorflow/blob/master/images/resnet-50.png. [Online; accessed 12-Oct-2017].

Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. *Cancer research*, 27(2):209–220.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

McDonald, D. M., Chen, H., and Schumaker, R. P. (2005). Transforming open-source documents to terror networks: The Arizona TerrorNet. In *American Association for Artificial Intelligence Conference Spring Symposia*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. pages 1–12.

Mittermayer, M. and Knolmayer, G. F. (2006). NewsCATS: A news categorization and trading system. pages 1002–1007. Data Mining, ICDM '06. Sixth International Conference.

Moat, H. S., Curme, C., Avakian, A., Kenett, D. Y., Stanley, H. E., and Preis, T. (2013). Quantifying Wikipedia Usage Patterns Before Stock Market Moves. *Scientific Reports*, 3:1–5.

Moschitti, A. (2006). Efficient convolution kernels for dependency and constituent syntactic trees. In *European Conference on Machine Learning*, pages 318–329. Springer.

Pavlidis, T. and Horowitz, S. L. (1974). Segmentation of plane curves. *Computers, IEEE Transactions on*, 100(8):860–870.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Qu, H. and Kazakov, D. (2016). Quantifying correlation between financial news and stocks. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6, Athen, Greece.

Qu, H., Sardelich, M., Qomariyah, N. N., and Kazakov, D. (2016). Integrating time series with social media data in an ontology for the modelling of extreme financial events. In *LREC 2016 Joint Second Workshop on Language and Ontology & Terminology and Knowledge Structures*, Portoroz, Slovenia.

Reimers, N. and Gurevych, I. (2016). Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks.

Robertson, C., Geva, S., and Wolff, R. (2007). Predicting the Short-term Market Reaction to Asset Specific News: Is Time Against Us? ... *in Knowledge Discovery and Data Mining*, pages 15–26.

Rubner, Y., Tomasi, C., and Guibas, L. J. (1998). A metric for distributions with applications to image databases. *Computer Vision, 1998. Sixth International Conference on*, pages 59–66.

Ruiz, E. J., Hristidis, V., Castillo, C., Gionis, A., and Jaimes, A. (2012). Correlating Financial Time Series with Micro-blogging Activity. *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining - WSDM '12*, page 513.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Sandoval, J. and Hernández, G. (2015). Computational visual analysis of the order book

dynamics for creating high-frequency foreign exchange trading strategies. *Procedia Computer Science*, 51(1):1593–1602.

Sandoval, J., Nino, J., Hernandez, G., and Cruz, A. (2016). Detecting informative patterns in financial market trends based on visual analysis. *Procedia Computer Science*, 80:752–761.

Schumaker, R. P. and Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Trans. Inf. Syst.*, 27(2):12:1–12:19.

Sedding, J. and Kazakov, D. (2004). Wordnet-based text document clustering. In Pallotta, V. and Todirascu, A., editors, *COLING 2004 3rd Workshop on Robust Methods in Analysis of Natural Language Data*, pages 104–113, Geneva, Switzerland.

Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., and Deng, X. (2013). Exploiting Topic based Twitter Sentiment for Stock Prediction. In *ACL (2)*, pages 24–29.

Si, J., Mukherjee, A., Liu, B., Pan, S. J. S., Li, Q., and Li, H. (2014). Exploiting Social Relations and Sentiment for Stock Prediction. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1139–1145.

Sokal, R. R. (1979). Testing Statistical Significance of Geographic Variation Patterns. *Systematic Zoology*, 28(2):227–232.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Tetlock, P. C. (2007). Giving Content to Investor Sentiment : The Role of Media in the Stock Market Published by : Wiley for the American Finance Association Stable URL : http://www.jstor.org/stable/4622297 The Role of Media in the Stock Market Giving Content to Investor Sentim. 62(3):1139–1168.

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Thomas, J. D. and Routledge, B. (2003). *News and Trading Rules*. PhD thesis, Carnegie Mellon University.

Wang, B., Huang, H., and Wang, X. (2012). A novel text mining approach to financial time series forecasting. *Neurocomputing*, 83(0):136–145.

Wolfram, M. S. A. (2010). *Modelling the Stock Market using Twitter*. Master thesis, The University of Edinburgh.

Zhai, Y., Hsu, A., and Halgamuge, S. (2007). Combining news and technical indicators in daily stock price trends prediction. In Liu, D., Fei, S., Hou, Z., Zhang, H., and Sun, C., editors, *Advances in Neural Networks ISNN 2007*, volume 4493 of *Lecture Notes in Computer Science*, pages 1087–1096. Springer Berlin Heidelberg.

Zhao, J., Lan, M., and Tian, J. F. (2015). ECNU : Using Traditional Similarity Measurements and Word Embedding for Semantic Textual Similarity Estimation. *Proceedings of the First Workshop on Social Media Analytics - SOMA '10*, (SemEval):117–122.