Socio-Affective Computing 9

Frank Xing
Erik Cambria
Roy Welsch

Intelligent Asset Management

Springer

Socio-Affective Computing

Volume 9

Series Editors
Amir Hussain, University of Stirling, Stirling, UK
Erik Cambria, Nanyang Technological University, Singapore, Singapore

This exciting Book Series aims to publish state-of-the-art research on socially intelligent, affective and multimodal human-machine interaction and systems. It will emphasize the role of affect in social interactions and the humanistic side of affective computing by promoting publications at the cross-roads between engineering and human sciences (including biological, social and cultural aspects of human life). Three broad domains of social and affective computing will be covered by the book series: (1) social computing, (2) affective computing, and (3) interplay of the first two domains (for example, augmenting social interaction through affective computing). Examples of the first domain will include but not limited to: all types of social interactions that contribute to the meaning, interest and richness of our daily life, for example, information produced by a group of people used to provide or enhance the functioning of a system. Examples of the second domain will include, but not limited to: computational and psychological models of emotions, bodily manifestations of affect (facial expressions, posture, behavior, physiology), and affective interfaces and applications (dialogue systems, games, learning etc.). This series will publish works of the highest quality that advance the understanding and practical application of social and affective computing techniques. Research monographs, introductory and advanced level textbooks, volume editions and proceedings will be considered.

More information about this series at http://www.springer.com/series/13199

Frank Xing • Erik Cambria • Roy Welsch

Intelligent Asset Management

Springer

Frank Xing
School of Computer Science and Engineering
Nanyang Technological University
Singapore, Singapore

Erik Cambria
School of Computer Science and Engineering
Nanyang Technological University
Singapore, Singapore

Roy Welsch
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA, USA

ISSN 2509-5706 ISSN 2509-5714 (electronic)
Socio-Affective Computing
ISBN 978-3-030-30262-7 ISBN 978-3-030-30263-4 (eBook)
https://doi.org/10.1007/978-3-030-30263-4

Foreword

I am fortunate to be a witness for the completion of this book. I came to know Dr. Frank Xing around 2 years ago when he approached me to seek collaboration. At that time, both of us were interested in working on using natural language processing techniques for solving problems in the stock market. This is a highly interdisciplinary area where knowledge in both finance and artificial intelligence would be useful. When we were introduced, I had been working on event-driven stock prediction for some time, having proposed some seminal work by leveraging deep learning algorithms. In parallel, Frank was fascinated about introducing sentiment signals to the forecasting of stock price movements.

We turned out to have a very pleasant collaboration project, learning a lot from each other. Frank has given me much inspiration from his passion and unique background with both financial and computational linguistics expertise, which has benefited from his interdisciplinary studies from Peking University and his devoted self-education in his doctoral research. As a computer scientist, I have gained more insight into the problem of financial market prediction that considers more than one asset by interaction with him. By the time he finished his doctoral research, Frank has become a leading expert in the field of intelligent asset management.

The application of artificial intelligence in the financial field has generated a lot of excitement in the past few years. Many financial institutions are facing real-world problems that are very close to what Frank discussed in this book. For example, Rebellion Research is a forefront quantitative asset management company that uses machine learning to invest in global equity. It launched its first pure AI investment fund in 2007. Based on machine learning, combined with predictive algorithms, and the support of decades of historical data, the company's trading strategy consistently outperforms in stocks, bonds, commodities, and foreign exchange transactions. Also, Bridgewater Associates, the world's largest hedge fund, established a new AI team in 2013 to automatically learn about market changes through probabilistic models. Similar companies include Point72 Asset Management, Renaissance Technologies, Two Sigma, and more. In terms of including sentiment signals, Sentient Technologies, a company founded in 2008, developed its first practical application in financial trading.

Many scholars would agree that apart from machine learning, the market has a growing interest in natural language processing techniques. The information obtained from historical data is very limited, and exploitation of textual data such as news, policies, and social media posts is powerful. In 2015, I proposed an event-driven method for stock market prediction using deep learning and natural language processing jointly. The events are extracted from news text and represented as dense vectors, trained using a novel neural tensor network, and then a deep convolutional neural network is used to model the combined influence of long-term events and short-term events on stock price movements. In simulations, a simple greedy strategy allowed our model to yield effective performance on the S&P 500 index prediction and individual stock prediction.

AI technology is evolving faster than expected and is already taking over human decision-making in certain instances. While many are alarmed by this, AI is producing some of the most effective and dramatic results in business today. This book, written in collaboration with my colleague Professor Erik Cambria and Professor Roy Welsch, has been a much-extended version of Frank's doctoral thesis. It is based on two well-established asset management models in the finance literature, injecting various AI approaches for better linkage between financial texts and market models. It provides a succinct and useful review of asset allocation models and then introduces the basis of how to model financial texts. In particular, computational semantic representations and texts, as well as sentiment knowledge encodings, are discussed. Finally, recent advances in knowledge engineering and dialogue techniques are discussed with regard to asset management. Theoretical introductions are accompanied by empirical results, which make the content of the book more practically informative.

I enjoyed reading this book much. Compared to the plethora of materials on intelligent stock trading, the book is unique in the following aspects. First, it provides a solid framework for asset allocation, considering expected returns and asset correlations in a unified base. This is different from most of the automatic trading algorithms in the literature, which have an overly simplistic model of asset allocation. Second, it gives much background on natural language processing, and in particular sentiment analysis, which is highly relevant to market prediction but typically oversimplified in the financial literature. Thus, I find this book a dedicated discussion of the cutting-edge techniques on intelligent asset management, which can be a useful reference for both academic research and industrial practice.

School of Engineering, Westlake University Yue Zhang
Hangzhou, China
April 2019

Preface

The scenario when investors need to manage a large number of financial assets has an essential difference from what most of the people do for stock movement prediction today. Unlike the situation of considering a single stock, investors need to consider co-movement of related stocks and control risk within a certain level. In traditional asset allocation models, expected returns and correlations of financial assets are difficult to estimate from historical price series, which are nonstationary and volatile. Therefore, we resort to textual knowledge hidden behind the huge amount of unstructured market information produced by human beings. In fact, one of the central research topics of this book include incorporating natural language processing techniques into several asset allocation models and finding the proper variables in financial models that naturally link to the contents of financial reports and the market sentiment.

New perspectives investigated in the book extend the current framework of the Markowitz model and the Black-Litterman model by re-thinking asset expected returns and asset correlations. Instead of relying on the price series themselves, external information can be used. We try to inject into these two concepts new connotations—asset expected returns and asset correlations not in terms of numerical calculation but in terms of what we *know* about the assets. Both sub-symbolic AI and symbolic AI approaches are explored in this book, for semantic linkage and market view modeling, which are associated with key variables in asset allocation models.

In the introductory chapter, types of financial texts are reviewed and categorized. However, most of the existing approaches in financial text mining treat heterogeneous information sources with no difference at the current stage. We propose here, as a value-adding step, to separately consider semantics conveyed in financial texts and the sentiment time series formulated from social media posts. We also introduce to the readers basic concepts of asset management.

Afterward, recent advances in computational semantic representation of words (word2vec) and documents (doc2vec) are leveraged to construct a dependence structure of financial assets. This dependence structure (termed vine dependence) is known to be useful in robust estimation of the covariance matrix of asset returns,

which is a critical risk indicator of the asset combination held by investors. As our main contributions, a vine-growing algorithm is proposed, and a large empirical vine structure for main US stocks is constructed. The readers will benefit a lot from this original research and step-by-step explanations and may apply this method in their own asset management models and practices.

Furthermore, we study adding the market sentiment to infer the posterior distributions of asset expected returns. Specially, augmented sentic computing, a concept-level sentiment analysis method that takes advantages of syntactic features, is used in processing short Internet texts and forming mass opinion streams. A novel recurrent neural network design termed ECM-LSTM is used to transform market sentiment to subjective investor views and benchmarked with popular neural network architectures, such as DENFIS and LSTM, and linear forecasting models, such as ARIMA and the Holt-Winters methods. The sentiment views enable explaining asset reallocation decisions in a storytelling manner. In the end, optimizing the polarity scores in a sentiment knowledge base is discussed.

Another important feature of the book is that a series of experiments were conducted to test the simulated portfolio performances, the validity of sentiment time series, and the model scalability. We describe the experiments in much detail so that the methods are convincing and well-supported with data. We find the robust estimation of asset correlations by semantic linkages to be superior to estimation using historical price data in a sense that with the help of a proper semantic vine, the portfolio outperformed 80% to 90% of its peers (arbitrary vines) in terms of annualized return. The improvement in annualized return is circa 2% for incorporating sentiment and more than 10% for employing ECM-LSTM compared to those fundamental settings.

Finally, we discuss storage and adaptation of knowledge and robo-advisory. This part is not directly related to the asset allocation models but an indispensable infrastructure to facilitate the model accuracy and human-computer interaction processes. To the end users, robo-advisory may be the only observable image and what all it means with the term "intelligent asset management." We hope that this book will increase readers' understanding of how to systematically integrate textual knowledge and market sentiment for financial asset management and incentivize researchers, policy-makers, professors, and entrepreneurs as a useful handbook.

Singapore, Singapore — Frank Xing
Singapore, Singapore — Erik Cambria
Cambridge, MA, USA — Roy Welsch
February 2019

Acknowledgments

The main contents of this monograph are extended and orchestrated from my doctoral research work, which would not have been accomplished without the help of many people. My foremost gratitude goes to Erik Cambria, who gave me sufficient independence as well as guidance on my research. Erik is not only a great supervisor but also a reliable old friend. He knows the oriental humility and often says "I did nothing for it," which, of course, is not true. Without his encouragement, I may never think of systematically organizing such a lot of material into a book. I am also deeply indebted to Roy Welsch, for he opened the door to robust statistics for me. I would cherish the discussions we had when he, getting on in years, after a long flight, arrived in Singapore.

I would like to thank Biya Wu for his help in providing perspectives from the finance and investing industries, Okan Duru for sharing his knowledge and expertise in forecasting, Xiaomei for helping me with some experiments, and Lorenzo for our beneficial discussions on deep learning and the unforgettable trip to Java. Besides, I am really proud of and thankful to Filippo for he started without much background but implemented and added a great deal to my ideas.

I also would like to thank Professor Chai Quek and Sundaram Suresh for their helpful comments during the years of my postgraduate study at Nanyang Technological University, where I was mainly supported by a scholarship from Temasek Laboratories. Professor Doug Maskell, Anupam, Weichen from my oral defense panel, and anonymous examiners also gave useful suggestions on an early draft. I am grateful to Professor Weihong Huang for the sound training I received from his mathematical economics course.

The journey of doing multidisciplinary research is tough and solitary, but I am very fortunate to have had encouragement from many friends and comrades on campus, outside campus, and even back in China: Harry Xia, Sipiao, Rui Yin, Te Bao, Cláudia, Soujanya, Yukun, Yang Li, Chen Qian, and others. I am grateful for Win-Bin and Yang Xu who gave me sound advice and still keep in touch after my graduation from Peking University. A special thanks to Jennifer for checking and pointing out an error in algorithms, and of course, I am responsible for the rest of the errors that are not found.

Most dearly, thanks to my Jessie for visiting Singapore very often for me and pull me through difficult times (even before our marriage). Above all, I would like to thank my family, especially my mama, Zhifei, for her everlasting and unconditional support.

Singapore, Singapore Frank Xing
June 2019

Contents

List of Figures

List of Tables

Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
ASCII	American Standard Code for Information Interchange
BOW	Bag-of-Words
CAGR	Compound Annual Growth Rate
CAPM	Capital Asset Pricing Model
CDAHS	Cognitive-Inspired Domain Adaptation with Higher-Level Supervision
CFA	Chartered Financial Analyst
CLSA	Concept-Level Sentiment Analysis
CPU	Central Processing Unit
DENFIS	Dynamic Evolving Neural-Fuzzy Inference System
DNN	Deep Neural Network
ECM	Evolving Clustering Method
EMH	Efficient-Market Hypothesis
ETF	Exchange-Traded Fund
EW	Equal-Weighted Portfolio
FNN	Fuzzy Neural Network
GARCH	Generalized Autoregressive Conditional Heteroscedasticity
GECKA	Game Engine for Commonsense Knowledge Acquisition
GICS	Global Industry Classification Standard
GMRAE	Geometric Mean Relative Absolute Error
JSON	JavaScript Object Notation
LSTM	Long Short-Term Memory
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
MDD	Maximum Drawdown
MPT	Modern Portfolio Theory
MVO	Mean-Variance Optimization
NLFF	Natural Language Based Financial Forecasting
NLP	Natural Language Processing

NLTK	Natural Language Toolkit
NT	Neural Trading Portfolio
NYSE	The New York Stock Exchange
OMCS	Open Mind Common Sense
PDF	Probability Density Function
PMI	Pointwise Mutual Information
POS	Part of Speech
POMS	Profile of Mood States
RBM	Restricted Boltzmann Machine
RDF	Resource Description Framework
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RSS^1	Really Simple Syndication
RSS^2	Residual Sum of Squares
SGD	Stochastic Gradient Descent
SVM	Support Vector Machines
SWF	Sovereign Wealth Funds
TF-IDF	Term Frequency-Inverse Document Frequency
TRBC	Thomson Reuters Business Classification
URL	Uniform Resource Locator
VW	Value-Weighted Portfolio
WNA	WordNet-Affect
XBRL	eXtensible Business Reporting Language

Symbols

a	financial asset
b	bias (perceptron)
C	capital amount
$\mathscr{C}$	clustering centroid
D	a document or dimension
$\mathbb{E}$	expectation
$\mathscr{E}$	edge (graphical model)
$\mathbb{F}$	function approximator
G	Gaussian function
$\mathbb{I}$	identity matrix
$\mathbf{I}$	conditional frequency
ℓ	loss function
L	lag operator
$\mathscr{L}$	sentiment lexicon
M	iteration times
$\mathscr{O}$	time complexity
P	asset mentioning matrix
Q	subjective expected returns
r	probability distribution for returns
R	asset return
$\mathbb{R}$	the set of real numbers
s	cosine similarity
S	semantic linkage matrix
$\mathbb{S}$	sentiment information
T	a tree or a record in training dataset
U	neural network parameters
v	trading volume
$\mathscr{V}$	a vine structure
w	portfolio weights of asset
w^*	optimized portfolio weights
W	state transition matrix

x	a word or a concept
X	explanatory variable
y	sentiment label
Y	response variable
$\mathbb{Z}$	the set of integers
α	system performance
β	volatility measure
γ	polarity score
δ	risk aversion indicator
ϵ	white noise
ζ	heuristic search range
η	desired system performance
θ	threshold
μ	expected asset return
π	asset price
ϖ	semantic partial correlation
$\wp$	semantic partial correlation matrix
Π	equilibrium risk premium
ρ	partial correlation
σ_{ij}	covariance between two assets
Σ	covariance matrix
τ	confidence level of CAPM
ϕ	autoregressive coefficients
ψ	augmented regression coefficients
Ω	view confidence matrix

Chapter 1
Introduction

All models are wrong, but some are useful.
— George E. P. Box

Abstract This introductory chapter revisits the historical progress of financial news analytics. In particular, the chapter emphasizes the importance and necessity of having asset allocation models for automatic asset management, superseding the first wave of predicting individual asset prices. We explain the development stages of natural language-based financial forecasting and summarize the scientific contributions of this book. Our extension brings new features to the current asset allocation models, such as transparency, flexibility, and robustness. The organization of chapters is provided as a roadmap at the end.

Keywords Artificial intelligence · Natural language based financial forecasting · Fin-tech · Asset management · Sentiment analysis

The recent boom of artificial intelligence (AI) has influenced many other fields, revolutionized the modes of thinking, and created interdisciplinary areas such as computational finance, contract review, and e-health. Stock market prediction is probably the most intriguing part of finance that draws comprehensive attention. Although many people simply define it as a classification or regression problem and build the model to predict prices from scratch, there has already been a number of theories and results on stock returns in the past decades [54, 108, 112]. It would be a pity to overlook those results, and the most straightforward approach can sometimes be naïve. For instance, since it is impossible to predict future prices with a very high accuracy from a very noisy background, it remains unclear how to compare two methods with different levels of risk. Moreover, practitioners are faced with multiple assets and instruments in the market. The price prediction paradigm also never elaborates how to incorporate multiple forecasting outcomes and decide holding positions. The asset allocation paradigm, therefore, is more powerful for market modeling.

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4_1

This book focuses on the asset allocation problem and aims to address a central question: how to leverage natural language processing (NLP) techniques to strengthen asset allocation models. We call this process textual knowledge integration, because it concerns an ecosystem comprising not only text mining but also curating and updating of knowledge. In this chapter, we first provide the background of natural language based financial forecasting (NLFF) [190] and our motivation for paying special attention to the asset allocation problem. Furthermore, we explain the goals of this book. Finally, we summarize our scientific contributions and describe the structure of the book.

1.1 Background and Motivation

Computer usage in finance industries has a longer history than many people would imagine. In the early 1950s, IBM began manufacturing proof machines that help with process automation.[1] Afterward, investment science also automated some of its numerical computation. However, the idea of using a computer to process human language was limited to academia for many years. Although reading and utilizing textual data to improve our understanding of the financial market dynamics has long been the tradition of trading practice, there were prevailing doubts on whether this professional practice can be automated, e.g., the Turing test [173].

Approaches that try to dispense with natural languages were developed by econometricians parallel to expert analysis. Numerical financial data are more accessible, because they are carefully curated since the establishment of financial markets. However, this resource is exhausting faster than many people imagined. Interestingly, apart from econometricians' increasingly complicated pattern mining models, market prediction that solely explores historical data seems more and more difficult. According to the analysis of [137] using the Hurst exponent,[2] the correlation between Dow Jones daily returns and its historical data receded from the 1990s. This result casts a shadow on the effectiveness of a group of autoregressive models.

Looking back at the nonnumerical data, the growing volume of financial reports, press releases, and news articles again galvanizes the wish to run the investment analysis automatically to keep a competitive business advantage. The earliest attempts to import other predictors employed discourse analysis techniques developed from linguistics [61] and naïve statistical methods such as word-spotting [23]. However, many challenges are unsolved at that time for the idea of automatically analyzing textual information. For example, the most popular way of representing sentences and paragraphs was bag-of-words (BOW), which may not be adequate to the task of comprehensive or deep understanding because the context information

[1] http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/bankauto

[2] The Hurst exponent is an index of long-range dependence that measures the rate at which the autocorrelations of the time series decrease as the lag between pairs of values increases.

get lost; the paradigm of knowledge engineering research also bounds the focus on a small portion of highly structured texts, while financial texts cover much broader topics. The construction of domain ontologies or semantic networks relies on very reliable and noise-free materials, for this reason, information about corporations from Internet Stock Message Boards and forum discussions [3] (recently often referred as "alternative information sources") were seldom considered.

In the first decade of the twenty-first century, the standard financial news analyzing system usually involved a mixed collection of news articles and stock quotes, as described in [148]. News articles are represented with their concatenated statistical feature vectors, for instance, word frequencies together with a one-hot representation of key noun phrases and name entities. Popular machine learning algorithms at that time, usually Support Vector Machines (SVM) [62] or evolutionary heuristics [22], are applied to blend the vector feature together with numerical data, to predict stock movements.

From 2010 onward comes the big data era. Social media websites such as Twitter, Facebook, and professional platforms such as StockTwits, eToro, etc. have generated an exponentially increasing amount of user content. The news analytics community once developed a special interest in mining this real-time information for mass opinions. Numerous papers especially pore over Twitter contents because of the relatively simple semantics conveyed in a restricted 140 character length [19, 154, 181]. Besides of the enrichment in different types of text sources, in this stage, more sophisticated NLP techniques are proposed. Sentiment analysis resources, such as Opinion Lexicon [76], are proposed; topic model [15] is used to discover both aspect and the related sentiment [122]. Machine Learning methods and knowledge-based techniques are simultaneously used for sentiment analysis as a core component. Neural networks, including a myriad of deep learning variants like convolutional neural networks (CNN) [49], restricted Boltzmann machines (RBM) [198], long short-term memory (LSTM) networks [94], etc., are experimented with prediction algorithms. Sometimes these models are also applied together with classic time series models such as autoregressive integrated moving average models (ARIMA) [99, 199].

Stepping back for a holistic view, we are at the dawn of the semantics curve of NLP technologies [30]. NLP systems start to approach human understanding accuracy at the sentence level. For instance, on categorical classification tasks such as distinguishing positive and negative sentences and choosing the best answer for reading comprehension, AI has achieved human-level performance. Processing more complex sense groups on a larger scale is becoming promising. On the other hand, although many text mining techniques have been experimented with NLFF, most of the studies still regard financial assets as discrete, independent, and broken pieces. Just like the performance of NLP systems is fueled by the increasing capability to understand concepts, contexts, progression, and opposition, NLFF is empowered by mastering the relationship between different assets. Finally, with

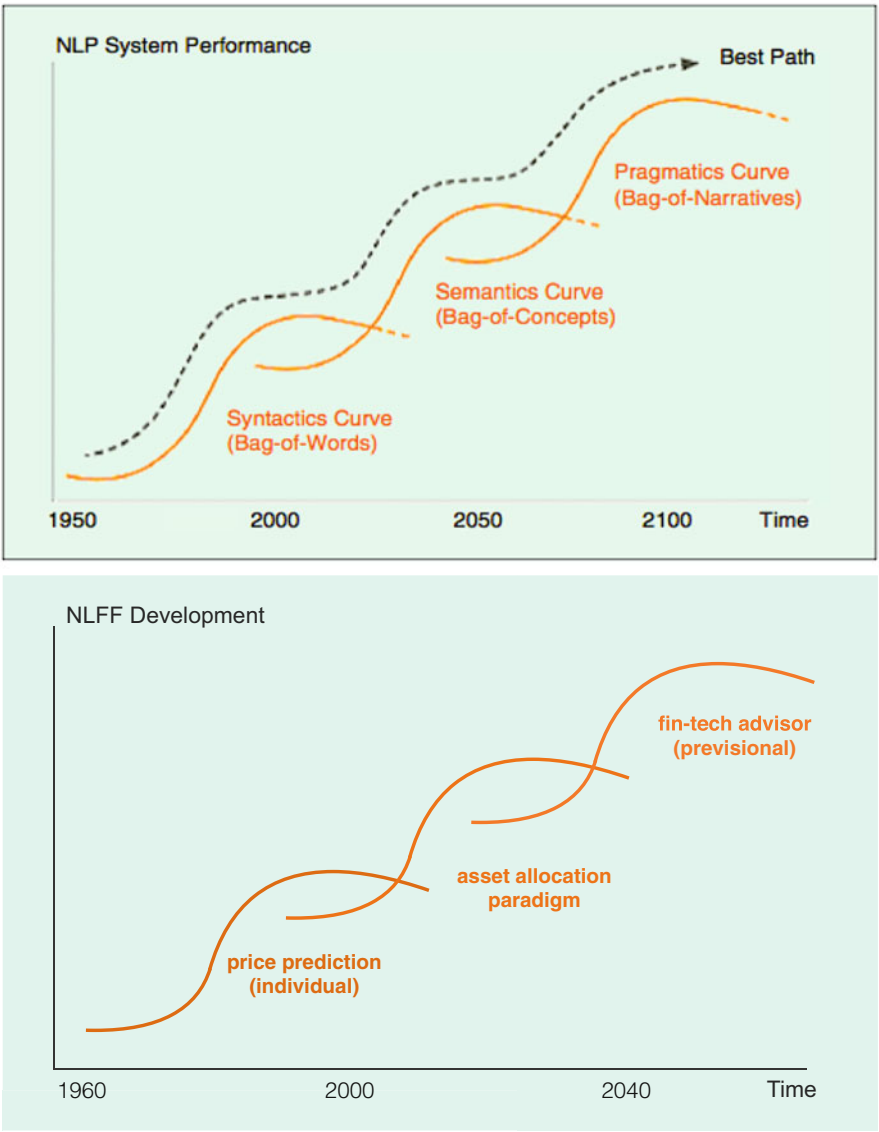

Fig. 1.1 Evolution of NLP techniques and NLFF waves

more peripheral knowledge, an artificial fin-tech expert would help clients with their lifelong financial plans, customize investment portfolios[3], and control risk (Fig. 1.1).

[3]Investment Portfolios refer to any combination of financial assets such as different stocks, bonds, cash, etc.

Our motivation is to automate the asset allocation process and to better understand what is going on when people make investment decisions. The current practice of analyzing fundamentals of specific companies can be dull and overwhelmed by the huge amount of information, while the underlying decision process remains in the dark. We have special interests in the following research questions. Are the posts people share on social media really consistent with what they think, especially on financial topics? Are the market prices driven by dominant opinions agreed to by the majority, or does truth always rest with the minority? What is the mechanism of interactions between market participants' opinions and their behaviors?

Asset allocation models are developed upon hypotheses, e.g., investors' risk tolerance is knowable a priori, and latent variables such as expected returns of assets do exist. However, little agreement has been achieved on how to estimate these variables. Most of the existing approaches employ past observations of these variables as the source of information, though detailed statistical method varies. As is known to many, the stock prices are non-stationary, and very volatile across time. The estimation of expected returns therefore is very sensitive to the choice of time span. Similar difficulties exist for estimation of correlations between different assets. People realize that relying solely on historical data is not a good idea. Besides behavioral finance, another way to explore these variables is via social experiments, which were considered expensive and impracticable in the past. Thanks to the advent of Web 2.0, large survey and opinion mining such as using the Amazon Mechanical Turk is becoming possible. We know the individuals in financial markets better than almost any time in history. This subsequently brings us new visions on how to allocate financial assets.

1.2 Objectives and Specifications

The research described in this book aims to provide an overview of the current approaches to text mining for financial forecasting—how financial indicators are formed and integrated to the operations that investors can take. In concrete terms, we investigate new perspectives to extend the current framework of asset allocation by rethinking asset expected returns (μ) and asset correlations (Σ).

More specifically, inspired by Bayesian asset allocation theories [4] and the Black-Litterman model [72], we extend the scalar representation of asset expected returns to a probabilistic distribution characterized by two parameters. As a result, the expected returns are accompanied by their confidence levels. This distribution is decided by fusing both the estimation from historical data and a subjective distribution inferred from the sentiment of market participants. Sentic computing methods are applied to analyze the user content from social media. This process provides the transparency required by most of the financial applications and enables quality assessment of the sentiment data stream.

Another extension we made is on the robust estimation of asset correlations. Instead of the historical price data, we use business descriptions to model the

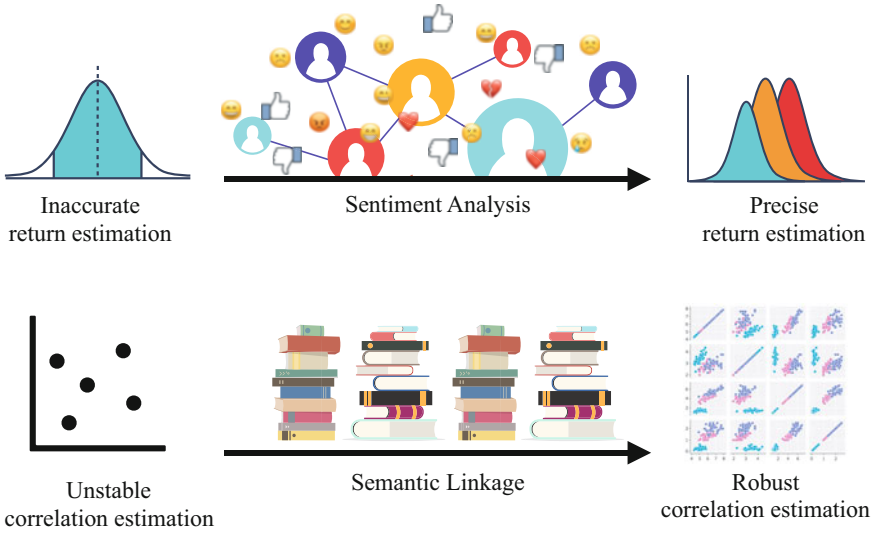

Fig. 1.2 Rethinking asset expected returns and asset correlations with two extensions

relation between company pairs. The target strength of linkage is estimated by similarities of various semantic representations of textual descriptions, which is less temporal-variant. A vine dependence structure is introduced to allow robustness of high-dimensional matrix estimation. This also saves computing power, because the traditional sliding window approach requires updates of correlation estimations for each time step of allocation. Figure 1.2 illustrates the two extensions.

Finally, the effectiveness of these two extensions depends on the hypothesis that NLP methods employed can capture the financial information as well as language nuance accurately. This cannot be achieved without domain-specific linguistic resources and tools. We investigated an algorithm to adapt general domain sentiment lexicon for the finance domain. This resource could be later applied to both aforementioned extensions.

1.3 Scientific Contributions

One of the main contributions of this book is blending scientific theories of asset returns and risks with the rapid development of NLP. Both communities will benefit from this research for it fills the gap between two narratives. Practitioners will find a more systematic guidance to apply sentiment analysis, name entity recognition (NER), and natural language representation for financial forecasting as the scope of computer-assisted asset management expands. Other contributions of this book are:

- The book rejuvenates knowledge-based approaches to asset allocation and portfolio management.
- Formalization of a Bayesian asset return estimator that incorporates mass opinion, linking sentiment and its adjustment to asset returns.
- Conceptualization of a semantic vine for financial assets, robust estimation of stock correlations based on a semantic vine.
- Proposal of a supervised algorithm that adapts sentiment lexicons to the target corpus domain, with potential financial applications.
- Extensive experiments and real-world applications to validate the efficacy of the above three contributions.
- Collection of textual financial data, including the over 80 MB Stocktwits® data stream spanning around 1 year (with user labeling), and over 50 business description articles from professional information vendors on major US stocks.

The remainder of this book is structured as follows. Chapter 2 is a retrospective chapter that surveys the text mining approaches and measurement people used for stock market prediction. The majority of them come from the computer science community and, therefore, belong to the first paradigm of NLFF waves. Preliminaries about asset allocation models are introduced as well. Chapter 3 discusses three different models of language structures and provides examples of leveraging semantics and sentiment in financial applications. Chapter 4 explains the recent advances in computational semantic representation of words and documents and how this representation could actually be used to construct a vine dependence structure for robust estimation of a covariance matrix of asset returns. Chapter 5 investigates the relation between market sentiment and the expected value of asset returns. The sentiment time series calculated from social media data streams are incorporated into the market sentiment views. Chapter 6 discusses storage form and update mechanism of semantic and sentiment knowledge. A sentiment adaptation algorithm is introduced to leverage user labels of sentiment from social media posts. Chapter 7 elaborates the key AI techniques that support the automation of the financial advisory process. Chapter 8 is a conclusive chapter that also includes limitations and future work. See Fig. 1.3 for the roadmap of this book. Finally, Appendices A and B provide some more details of the dataset used in this research and the data acquisition method.

The results and algorithms developed in this book have a direct impact on the asset allocation models and wealth management theories. Conventionally, quantitative method application is limited in this last fortress of finance as lots of uncertainties are involved, and customization is required to reflect investors' preference. By considering the shared knowledge and interaction with the virtual market participants, our framework shows several approaches that consistently improve the classic Markowitz's model and the state-of-the-art Black-Litterman model [72]. The book may have potential impacts on relevant fields of research, including NLP, machine learning, and econometric models. Researchers may realize the emergence of NLP techniques involved and advocate them for more application scenarios such as biomedical analysis, e-health, and education. These techniques are

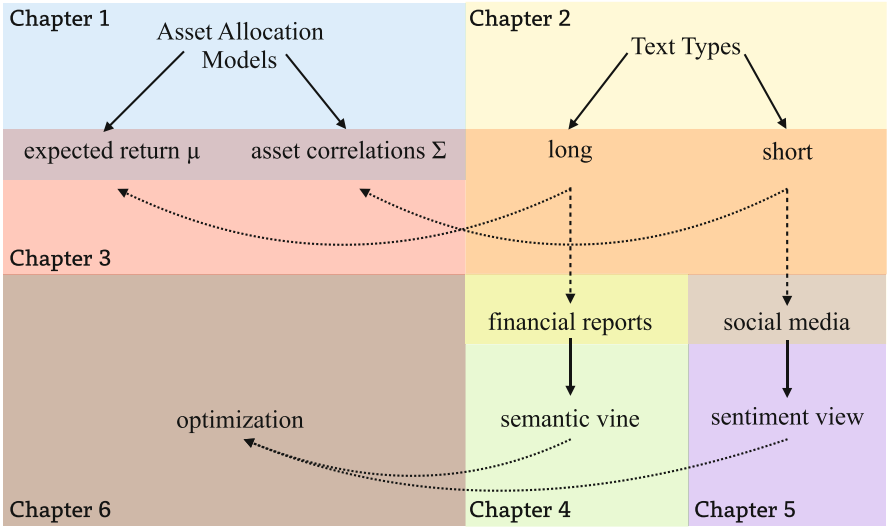

Fig. 1.3 The organization diagram of this book

representation learning, interpretable sentiment analysis, and domain adaptation. The book also leaves more questions unanswered, for instance, the theoretical aspect of ECM-LSTM, or more generally recurrent neural networks with filtering, is less explored. The machine learning community may find this type of network interesting, in spite of the famous attention mechanism [6]. Econometric models may also get inspiration by thinking about more complex nonlinear operations and stochastic optimization techniques rather than closed solutions.

Chapter 2
Literature Review and Preliminaries

Novelty emerges only with difficulty, manifested by resistance, against a background provided by expectation.
— Thomas Kuhn

Abstract This chapter reviews the text mining approaches employed and the problem formalization of stock market prediction by previous studies. A fine-grained categorization of text source is provided. The basic concepts and preliminaries of asset returns and portfolio optimization techniques are given in this chapter as well. The Markowitz model and the Black-Litterman model are the roots that connect financial variables with semantic modeling and sentiment analysis.

Keywords Stock market prediction · Text mining · Trading strategies · The Markowitz model · The Black-Litterman model

Creating an AI system equipped with investment guidelines and financial knowledge is not building castles in the air. We dedicate these ideas to the lost pearls [38, 171] that were never recognized as the mainstream in history. As the community appreciates data-intensive approaches more in recent days, it is even harder to realize the intellectual value of those early studies. Half a century ago, Clarkson [38] conjectured investment policies based on modeling of risk aversion of trust fund managers, and in fact, if they can really judge stock growth and income correctly, the chosen stock list would be at "the efficient frontier". The expert system (K-FOLIO) proposed by Trippi and Lee [171] is a good extension of the Markowitz model if the company grades integrated from its rulesets well-reflect the stock returns. If we have stepped further, it is by leveraging the flourish of opinion mining techniques and the wisdom of crowds.

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4_2

2.1 Text Mining for Stock Market Prediction

2.1.1 Text Source and Preprocessing

It has been noticed that there is a very diversified continuum of text sources from commercial to public, and the format, content, and authority can be systematically different. In a previous review article [190], we categorized the financial texts into six main groups according to three criteria: length of the texts, subjectivity level, and how frequently are they updated. The categories are listed below and examples are shown in Table 2.1.

Table 2.1 Financial texts from different sources and examples. (Partially adapted from [190])

Type	Characters	Example
Corporate disclosures	Long length, Subjective tone, Low frequency	Apple quarter reports: The company posted quarterly revenue of $84.3 billion, a decline of 5 percent from the year-ago quarter, and quarterly earnings per diluted share of $4.18, up to 7.5 percent. . . . . ."While it was disappointing to miss our revenue guidance, we manage Apple for the long term, and this quarter's results demonstrate that the underlying strength of our business runs deep and wide," said Tim Cook, Apple's CEO. . . . .
Financial reports	Long length, Objective tone, Low frequency	Quamnet portal: Gold prices went through a week of uncertainty due to mixed economic data. First there were weak retail sales data, which led gold prices to surge, yet investors remained uncertain how the data will affect the upcoming decision of the Federal Reserve. . . . .
Professional periodicals	Variable length, Objective tone, Mid frequency	Financial times: US consumers start to pay price of trade war with China. Economists fear households, and retailers face mounting burden as prospect of rising tariffs mounts. . . . .
Aggregated news	Mid length, Variable tone, Variable frequency	Yahoo! Finance: Tesla Inc. said on Thursday that it would roll out a software update to protect batteries while it conducts an investigation into incidents in which its vehicles caught fire. In a statement, Tesla said the software update will revise charge and thermal management settings on the company's Model S and Model X vehicles. . . . .
Message boards	Short length, Objective tone, High frequency	Amazon's board: The fact is. . . . . The value of the company increases because the leader (Bezos) is identified as a commodity with a vision for what the future may hold. He will now be a public figure until the day he dies. That is value
Social media	Short length, Subjective tone, High frequency	Twitter: Big turnaround in #AAPL, it is now over the 190 level up to 4 pts from morning lows

- Corporate disclosures are first-hand, reliable channels that the companies directly use to announce new information. The relation between price movement and corporate releases is self-reinforced as derived news relies on these materials. However, due to the lengthy nature and their relatively complicated structure, only a few studies managed to exploit this kind of source automatically with mixed news data. For example, Groth and Muntermann [68] investigate a collection of corporate disclosures required by the German security regulations.
- Financial reports are often written by market research institutions. Financial reports can have similar format as the corporate disclosures, but the content is more independent, reorganized according to themes, and verified by a third party. It is considered hard to maintain a balanced source of financial reports because they have to be individually accessed and have different standpoints. However, some research can still leverage financial reports because of their rigorous logic and high quality [32].
- Professional periodicals are the issued press of media companies printed regularly that have a special authority in finance, like *The Wall Street Journal* (WSJ), *Financial Times* [187], *Dow Jones News Services* (DJNS), *Thomson Reuters* [62], *Bloomberg* [49], and *Forbes* [140], to name a few. Most studies that we surveyed mix several from the abovementioned sources.
- Aggregated news, unlike professional periodicals which produce their own content, is a service that simply gathers the information from various professional periodicals. News Wire Services or news feeds (RSS) also belong to aggregated news. Some representative sources are Yahoo! Finance [88, 123, 148], Google Finance, and Thomson Reuter Eikon, which was formerly known as the product "TR3000 Extra" [62].
- Message boards are places that hold and store discussions like a forum. There is a directory of different topics, and market participants express their opinion under certain sections. Raging Bull [3], Yahoo's message board, and Amazon's message board [42] are message boards mentioned in the literature.
- Social media is an emerging and fast-developing source from which financial news and events can also be extracted. The majority of studies paid their attention to Twitter [19, 124, 154]: the econ-political influence of this platform becomes clearer as the US president starts to use it often. Another tool to monitor social media is Google Trend, for which further natural language processing is not required to obtain a time series with the help of a search engine [35]. Generally, social media is noisy and covers the general domain. Hence for practical use, one needs to filter the data by a list of financially related keywords. The advantage of social media content is that they can be used to monitor real-time information outbursts, though users need to bear the risk that their access may be deprecated.

The production process of the news also naturally brings about repetition and conflicts among different text sources; therefore, choosing the correct match for the task to be completed is vital for its success. Table 2.2 summarizes the specific information on what kinds of text sources are investigated and the way they are processed for previous studies in chronological order. We can observe that from

Table 2.2 Type of financial texts leveraged and how are they processed. (Partially adapted from [190])

Reference	Text type	Processing
Wuthrich et al. [187]	Professional periodicals	Manually crafted keyword tuples spotting
Lavrenko et al. [88]	Aggregated news	Alignment with trends
Fung et al. [62]	Professional periodical	Alignment with other stocks
Antweiler and Frank [3]	Message board	Naïve Bayes classifier
Das and Chen [42]	Message boards	Manually crafted sentiment lexicon
Tetlock et al. [168]	Professional periodical	Bag-of-negative-words
Schumaker and Chen [148]	Aggregated news	Bag-of-words, name entities, noun phrases
Bollen et al. [19]	Social media	Sentiment classification tool
Chan and Franklin [32]	Financial reports	Semantic class, instance-attribute pair
Groth and Muntermann [68]	Corporate disclosures	Risk model & indicator
Ruiz et al. [143]	Social media	Graph representation
Schumaker et al. [149]	Aggregated news	Pos/Neg & Sub/Obj classification
Si et al. [154]	Social media	Dirichlet processes mixture model
Si et al.[155]	Social media	Semantic stock network
Li et al. [96]	Mixed type	Emotion word dictionary
Ding et al. [49]	Professional periodicals	Neural tensor network (NTN)
Nofer and Hinz [124]	Social media	Sentiment classification tool
Nguyen et al. [123]	Message board	Latent Dirichlet allocation (LDA)
Yoshihara et al. [198]	Aggregated news	Recurrent neural network, RBMs

the starting point of this research field, professional periodicals are regarded as an important and primary text source. In preprocessing stages, filtering text source with a list of keywords or hashtags to a domain-specific or even company-specific materials rather than making use of the dataset as a whole with noise is a common practice. A more radical preprocessing in the case of "bag-of-negative-words" uses only negative keywords based on the belief that negative sentiment is more important in the context of financial forecasting.

Only in the past 5 years has the community started to gain increasing interest in social media. And because information disseminates at the second level, the data collected usually contain hundreds of thousands of messages. In such a circumstance, machine learning techniques are considered more often to output a sentiment index or summarization of the gist.

Processing or preprocessing of textual data is the procedure of preparing a well-formatted input. This input can be directly taken by a predictive model, which makes forecasts by running algorithms on the input. We roughly divided the popular text processing techniques into three groups.

The first group uses one-hot representation of keywords, keyword tuples, multi-word expressions, sentiment words, or more advanced statistics of them. Take market sentiment as an example; the percentage of positive mood on all relevant

word occurrences (sum of positive and negative mood states) is defined as "Social Mood Index (SMI)" by Nofer and Hinz [124]:

$$SMI = \frac{Positive\ Mood}{Grief + Hopelessness + Tiredness + Anger + Positive\ Mood}. \tag{2.1}$$

Zhang and Skiena [201] defined sentiment polarity as:

$$Polarity = \frac{positive\ count - negative\ count}{positive\ count + negative\ count}. \tag{2.2}$$

Similarly, a daily weighted mood word density time series in postings is defined as the optimism-pessimism mood scores (M_s^+ and M_s^-) by Li et al. [96].

The second group contains input formats suitable for specific machine learning algorithms, for instance, word embeddings [49], or probability distributions of that price moving up, moving down, or keeping steady condition on different words [3]. Yoshihara et al. [198] used a standard bag-of-words (BOW) model to represent the news articles. Although the temporal information of the articles are still preserved by utilizing a combination of recurrent neural network (RNN) and a restricted Boltzmann machine, the article representations obtained from the training phase were later incorporated to tune deep belief networks (DBN) that output the probability of an uptrend or a downtrend. The third group emphasizes to gather the alignments from texts to different trend motifs [88], triggers for related stocks, or simply the directional categories without further semantic or sentiment analysis of these alignments. In other words, this third group abandons the meaning of input format, and thus the learned model is similar to association rules between representations and the desired actions.

The text sources may not be always available for analysis. Although there had been multiple XML-format text sources distributed by the major financial information companies such as the Thomson Reuters News Feed Direct, Dow Jones Elementized News Feed, NASDAQ OMX Event-Driven Analytics, and Bloomberg Event-Driven Trading Feed. Probably affected by some commercial considerations, all these text data are no longer available as products. Instead, the raw data are transformed to more compressed formats, mostly from content vendors, that directly provide the processed sentiment data as a service. The most recently released products include Thomson Reuters MarketPsych Indices (TRMI),[1] RavenPack News Analytics, (RPNA)[2] PsychSignal,[3] YUKKA Trend,[4] TITAN[5], and so forth. The flagship product TRMI claims to cover a diversified range of text sources from

[1] http://https://www.marketpsych.com/

[2] http://https://www.ravenpack.com/page/ravenpack-news-analytics/

[3] http://psychsignal.com/

[4] http://www.yukkalab.com/yukkalab-news-trend/

[5] http://www.accern.com/

personal blogs to those major social media sites. However, the detailed data source list and how the texts are actually processed are not revealed. The similar situation goes for most of the commercial products mentioned above. Uhl [175] examined the correlation between the Thomson Reuters datastream and the corresponding stock returns qualitatively, but except for this, few studies attempted to evaluate the accuracy of these sentiment data.

2.1.2 Investigated Algorithms

In the past decades, linear regression and SVM have been classic yet dominant predictive models. Regression models are particularly preferred by econometricians because the model coefficients can be explicitly interpreted as the impact or elasticity of each factor included, and statistical tests are easy by comparing the current model with its alternatives by dropping out each variable. SVM is based on the sound Vapnik-Chervonenkis theory and has a key advantage that the support vectors which determine the hyperplane can be well-observed. Therefore, according to the recent survey by Kumar and Ravi [86], 70% of previous studies have adopted "regular" methods (decision trees, SVMs, etc.) and regression analysis. Our investigation roughly reconfirms this finding. Taking into account the nature of sparse, noisy, and unstructured financial data, excessively complicated models generally have a poor performance. However, one challenge in calibrating linear models is that they rely on strong hypotheses, for example, that errors are Gaussian and independently distributed. However, those hypotheses usually do not stand up for real-world problems. In spite of this, there are efforts to estimate some singular distributions that violate the strong hypotheses [9, 139, 166]; the outcome model is thus often bound to problems and cannot be generalized to different financial indicators. As beneficial supplementaries, neural network models and other statistical learning methods, such as Bayesian networks, are also prevalently experimented with.

In most of the studies, the features generated from the texts are not the only source of input. In fact, the features are combined with the numerical data, e.g., historical prices, to form an input datastream with richer information for prediction. In such cases, an ensemble method is required to fuse multiple models either on a feature level or a decision level. We are at a very primary stage to answer the question as to what category of algorithms is exclusively appropriate for the task of natural language-based financial forecasting [86]. Nevertheless, we can put popular algorithms of such kind into four categories: regressions, probabilistic inferences, neural networks and deep learning, or a hybrid of them.

Regression models have their special advantages in impact analysis. When the causality is clear, a linear regression can be enough and directly used with ordinary least square (OLS) or iteratively reweighted least squares (IRLS) to estimate coefficients [120]. Tetlock et al. [168], for instance, use a linear regression model

to illustrate that appearance of negative words in firm-specific news stories robustly predicts slightly lower returns on the following trading day. If we only consider prediction of directional market movements but not the intensity, the regression problem degenerates to a classification problem. SVM can naturally serve as a binary classifier and structured SVM for multiclass classification. Numerous pioneer studies indeed consider financial forecasting, or to be more specific, stock market prediction as a classification task. Support Vector Regression (SVR) [96, 148] is proposed to make discrete forecasting. Inspired by the idea that the empirical risk minimization objective can be used to build a regression model as well, SVR has the best from both SVM and linear regression. The hyperplane for SVR is also solved by deriving support vectors from the training data with a sensitivity threshold. A SVM calculates hinge loss for each wrongly classified data point. Unlike SVM, SVR gives more weight to data far away from the classification hyperplane due to the fact that this type of error would cause a huge loss in practice. Those data points close to the epsilon margin are not penalized. One drawback of SVR, however, is the requirement of introducing a kernel to project the training data into a linear separable higher dimension and an extra threshold parameter. These hyperparameters are often manually selected according to the features of data or without many fixed reasons. For instance, in the AZFinText system [148], the best performing SVR model combines both news articles terms (binary coded) and the baseline stock price, which is per se not linearly separable.

From many species of neural networks, Bollen et al. [19] select a self-organizing fuzzy neural network (SOFNN). The model is specially suitable for time series forecasting and regression problems. Compared to other fuzzy neural network (FNN) models, such as the most widely applied adaptive neuro-fuzzy inference system (ANFIS), SOFNN is faster due to its settings of how new membership function will be added. The topological structure of SOFNN is not different from other common fuzzy neural networks. However, the learning process is different and divided into two phases: structure learning and parameter learning. That is, in the first phase which is called "self-organizing learning," the number of fuzzy rules is fixed. After the establishment of neural network structure, the second phase only adjusts weight parameters of neurons, which is called "optimization-learning." Following the ideas of [148], Bollen et al. [19] simultaneously set the lagged Dow Jones Industrial Average (DJIA) value and generalized POMS (GPOMS) as two inputs of a SOFNN model. The output is a prediction of the current value of DJIA. Along with the recent advances in deep learning, neural networks can also be used to model the relationship between two entities by introducing an additional tensor to the entities. This modification increases the expressive power for text data. Ding et al. [49], for example, used a neural tensor network (NTN) to train event embeddings, which contains much more information than article terms and keywords. Later, a sequence of event embeddings with different time spans are fed into a convolutional neural network (CNN). The CNN model outputs a binary prediction of whether the stock price will move up or move down.

2.1.3 Assessment and Performance Measurement

Text mining for stock market prediction can be assessed *indirectly* through the reduced error in predicting (directional) price movement, the future prices [19, 148], or *directly* through a trading simulation [96, 143]. Error measures include precision, recall, F-score, and accuracy for binary or ternary classification and MAPE, GMRAE, RMSE, MASE, and DNME for forecast error [79, 172, 190]. However, it is difficult to know in the latter assessment whether the effectiveness should be attributed to trading strategy or text mining, given the fact that every research uses a slightly different trading strategy. Table 2.3 lists measurement and performance of previous studies. Note that some of them focus on descriptive research and do not provide any trading strategy.

For the rest of the studies, literature reported at least two threads of popular while unsophisticated trading strategies:

- Buy up/sell down: this strategy models the behavior of an investor. With daily rebalancing or other frequencies, the investor buys the stocks that will perform well according to his prediction or sell the stocks that will perform badly. Taking buy up as an example, the investor sells all the stocks at the closing time and holds the capital for the next round of investment. When the investor has multiple predictions, he will first calculate and rank the return ratio of stocks, and uniform split, or weight the "best-n" stocks. Ruiz et al. [143] used a very simple heuristic for the bin packing problem, which defines the weight for a stock as price difference over open price. The vague part is that why inferior stocks are still selected to diversify the portfolio.
- Short-term reversal: this strategy leverages the phenomenon that stocks with relatively low returns over the last period tend to earn positive abnormal returns in the following time period. Instead of relying on the prediction of future prices, the strategy focuses on nowcasting observations. For instance, if the sentiment for a specific stock is very pessimistic and returns are low at time t, the investor's nowcasting is that at time $t + 1$ the returns would "reverse." Consequently, the investor would like to buy in this stock at time t and hold long position through time $t + 1$ and vice versa [41]. Although critics argue against the hypothesis that reversal strategies require frequent trading and rebalancing in disproportionately high-cost securities, this strategy remains famous. In practice, though, stocks with relatively low returns over the last period are hard to trade and can have liquidity problems. So the strategy could lead to a situation that trading costs are higher than the normal standard and prevent profitable strategy execution. Lavrenko et al. [88] and Ding et al. [49] specified it as a timing practice. This makes it almost impossible for any theoretical analysis because the strategy executions will depend on the price trajectory. It seems that positions are completely determined by thresholds (hyperparameters). If we remove the timing mechanism, the strategy becomes very naïve and has some defects, e.g., assuming the investor can borrow an infinite amount of capital.

Table 2.3 Results achieved and reported using different measurements. (Partially adapted from [190])

Reference	Measurement & performance	Trading strategy
Wuthrich et al. [187]	Direction accuracy of five main indices, around 50%	Buy up/sell down, rebalancing daily
Lavrenko et al. [88]	Trading simulation of 127 stocks, average gain per transaction 0.23%	Short-term reversal, rebalancing hourly
Fung et al. [62]	Trading simulation of 33 stocks, cumulative profit 6.55%	Buy up/sell down, rebalancing daily
Antweiler and Frank [3]	Statistical testing for correlation with DJIA & DJII, significant predictor	Not mentioned
Das and Chen [42]	Statistical testing for correlation with MSH-35, correlation is weak	Not mentioned
Schumaker and Chen [148]	Closeness, direction accuracy, trading simulation, MSE 0.04261, Acc 57%, Return 2.06%	Not mentioned
Bollen et al. [19]	Closeness, direction accuracy for DJIA, MAPE reduction by 6%	Not mentioned
Chan and Franklin [32]	Even sequence correct accuracy, significant improvement (>7%)	Not mentioned
Groth and Muntermann [68]	Accuracy, precision, recall, option simulation, Acc 70%, p 47%, r 70%, many false positive	Not mentioned
Ruiz et al. [143]	Trading simulation on a 10-company portfolio, return of 0.32%	Buy up/sell down, rebalancing daily
Schumaker et al. [149]	Direction accuracy & trading simulation, Acc 59%, Return 3.30% (sub. news only)	Triggered short-term reversal
Si et al. [154]	Direction accuracy of S&P100 index, 68.0%	Not mentioned
Si et al. [155]	Direction accuracy on $AAPL, 78.0%	Not mentioned
Li et al. [96]	Closeness, direction accuracy, trading simulation, RMSE 0.63, Acc 54.21%, est. Return 4%	Short-term reversal
Ding et al. [49]	Direction accuracy, trading simulation, Acc 65.08%, Avg. Profit Ratio 1.679	Short-term reversal
Nofer and Hinz [124]	Statistic testing for correlation with DAX, trading simulation, AROR 84.96%	Buy up/sell down of ETF
Nguyen et al. [123]	Direction accuracy, Acc 54.41%	Not mentioned
Yoshihara et al. [198]	Direction accuracy, trading simulation, improved error rates and profit gain compared to SVM	Buy/sell at MACD turning point

These deficiencies impede comparing different approaches and more severely leave the accordingly constructed portfolios far from state of the art. Therefore, we propose to take a serious look at asset allocation models, which is an elephant in the room for stock market prediction in the wild. This paradigm shift from considering a single stock to managing many assets also helps to better understand properties of financial variables, which are more or less absent in the current paradigm of stock market prediction in computer science.

2.2 Asset Management

When talking about asset management, and even in the more specific context of financial asset management, there are many aspects to consider such as who is the asset owner, what is his/her objectives, and the accessibility to investment tools. Different asset owners, from Sovereign Wealth Funds (SWF) to trustees, university endowment, and individuals, have diversified level of tolerance to take risk. And exposure to (good) risk is actually the only way to be rewarded certain type of premiums. In different phases of a life cycle, asset owners' objective may change to meet the current needs: exploring opportunities or preparing for retirement. So it might be a bit oversimplified to stop at asset classes level—abstract the average expected returns and risk, and pass them to a calculator. At a practical level, one will also have to face asset liquidity problems, principal-agent problem in delegation, and make tax-efficient arrangements. Ang [2] provided a comprehensive discussion of these topics. In terms of intelligent asset management, these factors should all be naturally collected and considered.

There are at least three popular types of objectives for asset management:

- To meet requirements from the client, for instance, achieving certain levels of return. This objective can be formulated as a "goal programming" problem, and it is how artificial intelligence and operations researchers understand the purpose of asset management in the early days of the last century [90].
- To mitigate risk and generate return: this is the most widely acclaimed objective, especially after Markowitz developed the core idea of the modern portfolio theory (MPT). Though many issues arguably arise, such as misalignments between the incentives of the industry and its ultimate clients, short-termism [71], increased frictional cost, and the paradox of growing correlation between assets.
- To provide sustainable financial solutions, which is a combination of many personalized factors. Some realized the bad externality of thinking of asset management as a zero-sum game and systematic risk out of control, approaches that consider environmental and social impacts can eliminate fees and enjoy the first-mover advantage. This idea moves closer to general asset management; according to standard ISO 55000, the mixed objective contains political, social, and legal goals and more.

In the rest of this chapter, our focus will be on the second objective, where the most sophisticated theories are built.

2.3 Asset Allocation Models

Asset allocation models are rigorously defined implementations that consider not only stocks but a broader range of financial assets, including cash, fund, treasury bonds, foreign currency, derivatives, and many other tradable instruments on the

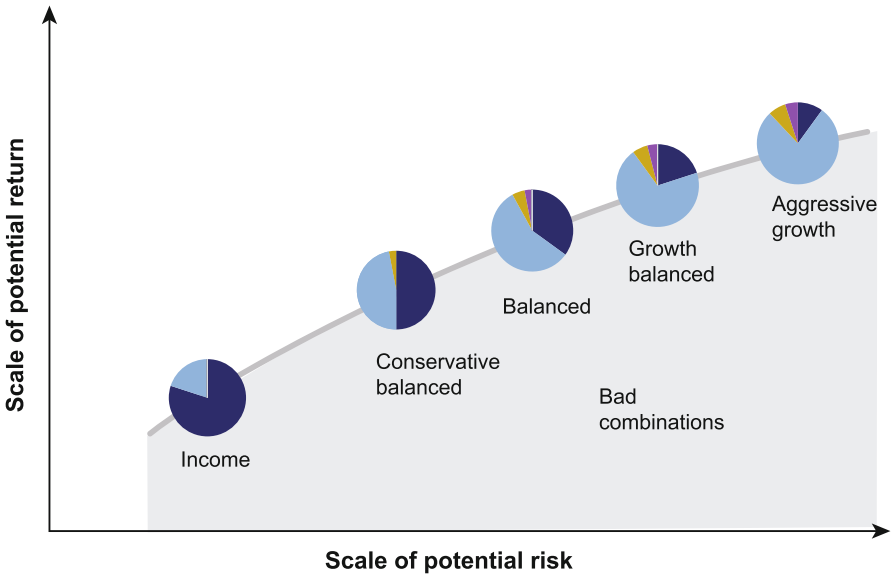

Fig. 2.1 Different asset allocation strategies

market. When people invest in a single asset, they have concern for two core issues: what is the expected return for the investment and how big is the risk. When the investor will have to share out a given amount of (initial) capital to multiple assets with different expected returns and risks, this problem becomes nontrivial because a bad combination achieves a lower potential return at a given risk level (see Fig. 2.1, adapted from [176]). This is because assets have their "idiosyncratic" risk. When the combination contains diversified and heterogeneous assets, some will zig while others zag, so to cancel out the overall fluctuation in the combination [71].

2.3.1 The Markowitz Model: Mean-Variance Portfolio

We introduce the widely acknowledged portfolio construction framework named after Harry Markowitz [108]. Consider a market with N assets. Define the percentage return[6] of an asset i as the increase in its value divided by its price per share:

$$R_i = \frac{\pi_{i,\,t+1} - \pi_{i,\,t}}{\pi_{i,\,t}} \,. \tag{2.3}$$

[6]Log returns are more frequently used in machine learning; however, percentage return is easier to understand for the purposes of portfolio optimization and risk management. The two forms are interconvertible and should be used in different environments.

Assume R_i is a random variable and its mean μ_i and variance σ_i^2 exist. Assume no other information is available and the investment is one-period.

If the investor decides to split his/her money to invest on multiple assets, then overall expected return of his holding will be the weighted average of expected returns of selected assets:

$$\mu_{\text{pfl}} = \mathbb{E}(R_{\text{pfl}}) = \sum_{i=1}^{N} \mu_i w_i \ , \tag{2.4}$$

where $0 \leq w_i \leq 1$ denotes the holding weights of each asset species i in the portfolio and μ_i denotes the expected return on asset i.

The idea of Markowitz's mean-variance method is that, since the expected return of a portfolio can be calculated as the mean of its component asset returns, the overall risk of a portfolio can be thus represented by the variance of portfolio expected return, which is already known:

$$\begin{aligned}
\sigma_{\text{pfl}}^2 &= \text{var}(R_{\text{pfl}}) \\
&= \mathbb{E}(R_{\text{pfl}} - \mu_{\text{pfl}})^2 \\
&= \mathbb{E}[\sum_{i=1}^{N}(R_i - \mu_i)w_i][\sum_{j=1}^{N}(R_j - \mu_j)w_j] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} w_i \sigma_{ij} w_j \\
&= \mathbf{w}' \Sigma \mathbf{w}
\end{aligned} \tag{2.5}$$

where σ_{ij} is the covariance between the returns on asset i and asset j and can be estimated from multiple observations. The holding weight variable $\mathbf{w}$ is a $N \times 1$ vector over the portfolio assets, Σ is a $N \times N$ (symmetric) covariance matrix where the element at the i-th row and j-th column is σ_{ij}. It is not difficult to understand the variance can be used as a simple measure of risk: the more "variable" of R_{pfl}, the larger σ_{pfl}^2 is. If the portfolio return R_{pfl} is certain, its variance is equal to zero, and so such a portfolio is called "risk-free" [53].

The investor wants to meet two objectives: (1) he wants to maximize the portfolio expected return, and (2) for a given return level, he wants to minimize the risk [108]. If we use the variance of portfolio return as a risk metric, we will have an optimization problem as follows: maximize $\mu_{\text{pfl}} - \frac{\delta}{2}\sigma_{\text{pfl}}^2$, where δ is an indicator of risk aversion or tolerance that trades off between the investor's two abovementioned objectives. More rigorously, without short selling or borrowing, the portfolio weights should be nonnegative normalized real numbers:

$$\max_{w_i} \sum_{i=1}^{N} \mu_i w_i - \frac{\delta}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_i \sigma_{ij} w_j \tag{2.6}$$

$$\text{s.t.} \sum_{i=1}^{N} w_i = 1,\ i = 1, 2, \ldots, N.$$

One good property of the optimization problem (2.6) is being quadratic concave regarding w_i. The optimized weights of an efficient portfolio are, therefore, given by the first-order condition of equation 2.6[7]:

$$\mathbf{w}^* = (\delta \Sigma)^{-1} \boldsymbol{\mu}. \tag{2.7}$$

where μ denotes a $N \times 1$ vector consisting of μ_i.

If we control the expected return of the whole portfolio to a fixed value e just like the "goal programming" problem setting, we will obtain sectional portfolio weight combinations. These combinations meet both the constraint on weight sum and the constraint on expected return. Under these constraints the optimization problem can be restated as:

$$\min_{w_i} \ \mathbf{w}' \Sigma \mathbf{w} \tag{2.8}$$

$$\text{s.t. } \mathbf{1}'\mathbf{w} = 1, \text{ and } \boldsymbol{\mu}'\mathbf{w} = e.$$

Figure 2.2 provides an illustration of the constraints visualization with a portfolio of three assets (see equation 2.8). The green plane denotes the weight constraint, and the cyan plane shows the expected return constraint. At the intersection of both planes is a segment, which is the feasible domain. For points on this segment, the heat map from black (minimum) to white (maximum) denotes the value range of $\mathbf{w}' \Sigma \mathbf{w}$. Note that the Markowitz model requires a positive-definite Σ. This means σ_i^2 is strictly positive and all the assets $i = 1, 2, \ldots, N$ are risky. Otherwise, there must exist some leading principal minor of Σ that is not strictly positive (Sylvester's criterion for symmetric real matrices). Actually, the covariance matrix geometrically defines the slope of planes in Fig. 2.2 and also guarantees the solution (2.7) exists and is unique [53].

We further illustrate how risk is mitigated with a higher expected return by diversification of the portfolio. Assume three financial assets with Gaussian distributed returns: $r(a_1) \sim \mathcal{N}(0.001, 0.02^2)$, $r(a_2) \sim \mathcal{N}(0.002, 0.05^2)$, and $r(a_3) \sim \mathcal{N}(0.003, 0.08^2)$. Since the covariance of two independent variables is zero, the covariance matrix has a form of:

[7]The derivation process is based on applying the method of Lagrange multipliers and Karush-Kuhn-Tucker conditions.

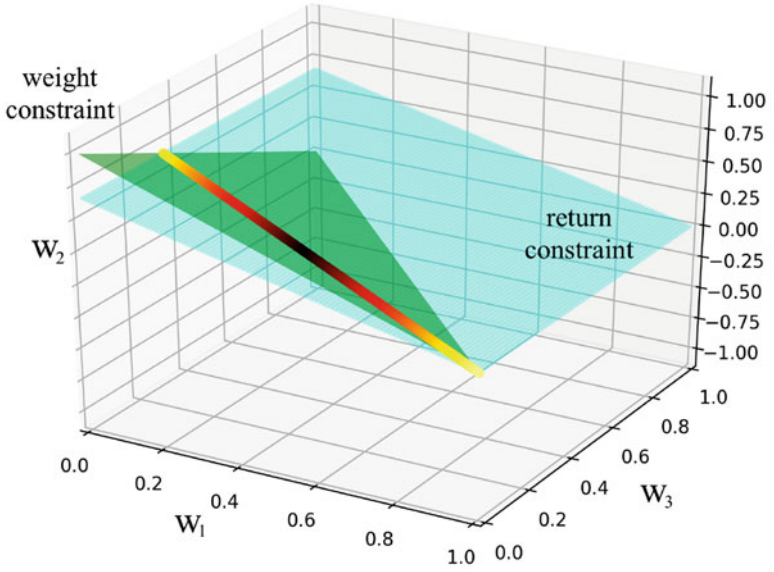

Fig. 2.2 A 3-D visualization of the portfolio optimization problem [191]

$$\Sigma = \begin{bmatrix} 0.02^2 & 0 & 0 \\ 0 & 0.05^2 & 0 \\ 0 & 0 & 0.08^2 \end{bmatrix},$$

According to Eqs. 2.4 and 2.5, if we hold a portfolio consisting of 20% of a_1, 50% of a_2, and 30% of a_3, expected return and variance of the portfolio would be:

$$\mu_{\text{pfl}} = 20\% \times 0.001 + 50\% \times 0.002 + 30\% \times 0.003 = 0.0021 \tag{2.9}$$

$$\sigma_{\text{pfl}}^2 = \sum_{i=1}^{3} \sigma_i^2 w_i^2 = 4\% \times 0.02^2 + 25\% \times 0.05^2 + 9\% \times 0.08^2 = 0.034^2. \tag{2.10}$$

This portfolio is strictly superior to holding only a_2 because it has a higher expected return ($0.0021 > 0.002$) and a lower risk ($0.034 < 0.05$). Figure 2.3 exhibits a simulation result showing the power of diversification.

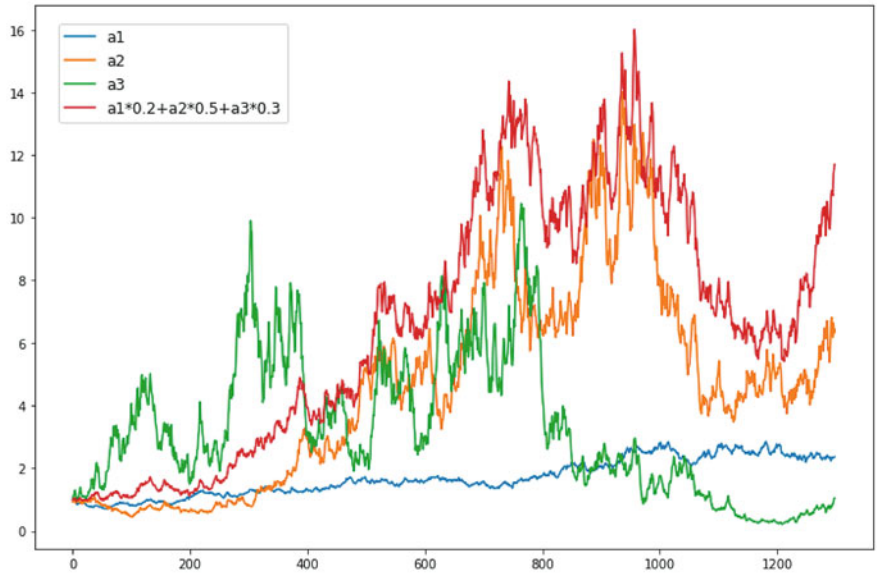

Fig. 2.3 The power of portfolio diversification

2.3.2 The Black-Litterman Model: Views on Market

Despite its theoretical elegance, many details have to be specified when applying Markowitz's mean-variance approach in real-world cases. For example, if we take a different number of observations of R_i, the calculated means tends to be significantly different. Generally speaking, the two moments of asset returns are extremely difficult to be estimated accurately from historical stock price series [153], as they are nonstationary and volatile.

Despite the fact that the Markowitz model suggests non-robust ways to estimate the expected asset returns and volatilities, the situation can be worsened. Because the Markowitz model itself is very sensitive to the inputs, that is, the estimated asset returns and volatilities. If a small error occurred in the estimation of μ or Σ, the final optimized weights outputs will have very different values, leading to very different rebalancing decisions. To address this error dispersion limitation of the Markowitz model, a Bayesian approach was proposed in [14]. The approach integrates the additional information of investor's judgment, which is usually from experts in finance, to the market fundamentals and the priors from the Markowitz model.

Traditional approaches to expected return estimation assume the equilibrium risk premiums are the same as that in the capital asset pricing model (CAPM). CAPM states that for certain asset, the expected return is higher because investing on such an asset is more risky. The model assumes that for one unit of risk, each asset requires for the same premium when the market is in equilibrium. Therefore, the equilibrium risk premium for asset i is proportional to the market premium:

$$\Pi_i = \mu_i - \mu_f = \beta_i(\mu_{\text{mkt}} - \mu_f) \tag{2.11}$$

where μ_{mkt} is the market expected return and μ_f is the risk-free interest rate, usually taken from the US Treasury Bond. β_i measures how volatile asset i is comparing to the market average. Equation 2.11 reveals that the estimation error in μ is further passed to Π. Generalized CAPM tackles this problem by allowing asymmetric and fat-tailed estimator of the return distribution [9], while the Black-Litterman model directly adjusts the expected return by allowing investor's subjective opinions.

The posterior expected returns in the Black-Litterman model μ_{BL} of the portfolio are inferred from two antecedents: the equilibrium risk premiums Π of the market as calculated by CAPM and a set of subjective views on the expected returns from a professional investor. The Black-Litterman model is based on probabilistic inference of equilibrium returns. We start with the simple assumption that the equilibrium returns are normally distributed with a mean equal to the risk premium and a variance proportional to the portfolio volatility, that is, $r_{\text{eq}} \sim \mathcal{N}(\Pi, \tau\Sigma)$. Note that Σ is the covariance matrix of asset returns as above elaborated and τ is an indicator of the confidence level of the CAPM estimation of Π, which links up volatilities of individual assets. To facilitate easy computation, the market views on the expected asset returns held by an investor are assumed to be normally distributed as well, denoted by $r_{\text{views}} \sim \mathcal{N}(Q, \Omega)$. We will explain the physical meanings of Q and Ω in later parts. This assumption helps to induce Gaussian posteriors.

If we denote the posterior distribution of the expected asset returns providing the CAPM model and the market views by r_{BL}, subsequently, it is mathematically clear that r_{BL} can also be written as a normal distribution characterized by two parameters: $r_{BL} \sim \mathcal{N}(\mu_{BL}, \Sigma_{BL})$, where both μ_{BL} and Σ_{BL} are dependent on the aforementioned variables (Fig. 2.4), that is:

$$[\mu_{BL}, \Sigma_{BL}] = \mathbb{F}(\tau, \Sigma, \Omega, \Pi, Q). \tag{2.12}$$

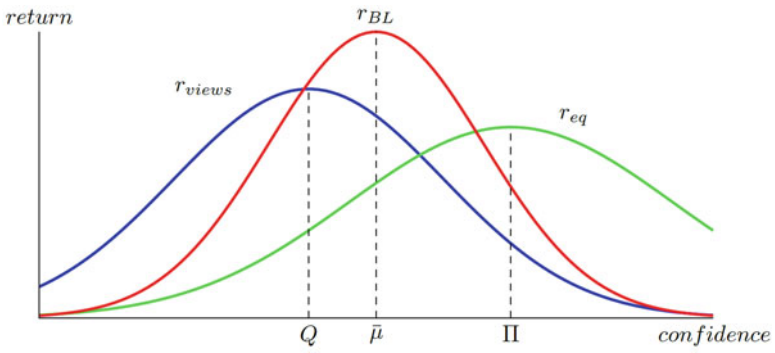

Fig. 2.4 Posterior distribution of expected returns as in the Black-Litterman model

We provide the analytic solution of Equation 2.12 based on Bayes' theorem:

$$pdf(\mu_{BL}) = \frac{pdf(\mu_{BL}|\Pi)\ pdf(\Pi)}{pdf(\Pi|\mu_{BL})} \tag{2.13}$$

With μ_{BL} and Σ_{BL}, the optimized Bayesian portfolio weights according to the mean-variance optimization framework will be similar to equation 2.7, only substituting the original variables Σ and μ by their Bayesian counterparts Σ_{BL} and μ_{BL}:

$$\mathbf{w}^*_{BL} = (\delta \Sigma_{BL})^{-1} \mu_{BL}. \tag{2.14}$$

The reason why the Black-Litterman model became so popular is that instead of explicitly giving the probability density function (*pdf*) of Bayesian posterior returns, it provides new interpretations to the probability distribution of expected returns. Since the distribution has a second-order expectation, understanding those parameters is not an easy work. By introducing the concept of market views, the Black-Litterman model describes one of the antecedents in a more natural and human-understandable manner. Because the views are subjective, this human-model interface using market views makes life easier for inserting investor's opinions. The definitions of market views are elaborated in Chap. 5.

The most common criticism on the Black-Litterman model is the same as what it extends the Markowitz model: the subjectivity of investor's views. However, since expected returns cannot be accurately estimated from historical returns, it is unfair to attack adjustments on them, which also brings more flexibility. The problem depends on how to obtain reliable and trustful views, which is the main topic of Chap. 5. To summarize, the Black-Litterman model makes substantial improvements on stabilizing the Markowitz model, though its performance heavily depends on the quality of the subjective market views. The Black-Litterman model leaves the question of how to resort to the finance experts and actually obtain these good quality views unanswered. This reality motivates us to think about possible solutions to automatically extract financial information as well as public mood on the Internet and form market views from such information.

Chapter 3
Theoretical Underpinnings on Text Mining

We should look at problems from different aspects, not from just one.

— Mao Zedong

Abstract This chapter provides multiple perspectives on the structure of natural language. We try to answer two fascinating questions in this chapter: what kind of information can we extract from human language, and is the extracted information sufficient or effective for financial forecasting? Three hierarchical representations of language and its functions are compared and aligned. We propose a dichotomy of semantics and sentiment underlying natural language, which is ideally suited for financial applications and takes into account facts about time. Finally, we present some examples to show that utilization of natural language in business areas has inadvertently followed this structure.

Keywords Language structure · Mental representation · Predictability · Grammar · Emotion · Financial information

3.1 Language and Its Fabrication

Language is an exclusive channel for human communication. Though some linguists would dispute, mature language ability is not found in other species. Therefore, the relation between language and human intelligence is believed to exist. Turing [173] proposed to judge intelligent behavior by natural language conversations, and today, NLP is a core subject of AI.

Cognitive science developed many hierarchical models to describe the emergence of intelligence, which is called *the computational theory of mind*. For instance, Pinker [132] discussed four functional aspects of intelligence: perception, reasoning, emotion, and society. These aspects comply with a psychological pyramid. The most fundamental computational intelligence, including memory and calculation, serves the base of the pyramid; perceptive intelligence, including visual, auditory,

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4_3

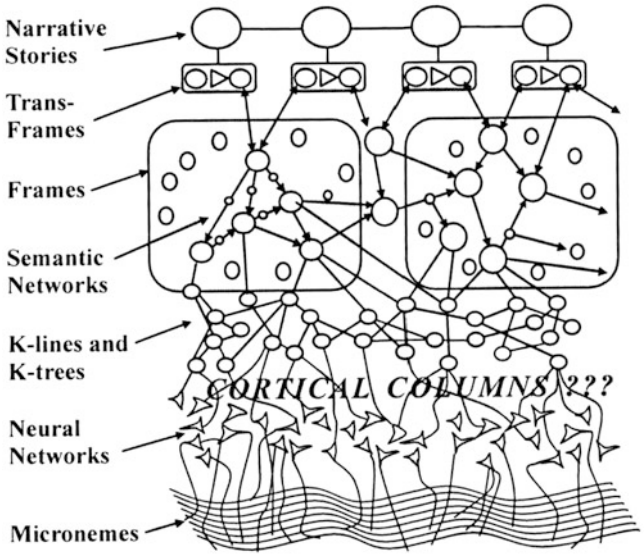

Fig. 3.1 A hierarchy of various types of knowledge [116]

and tactile reactions serves the second layer; cognitive intelligence, i.e., language, knowledge, and reasoning, forms the third; and at the top is creativity, self-consciousness, etc.

The same hierarchical structure also exists inside the mental representations of language. We enumerate two models. In the first one, Minsky [116] even extends the strata to biological structures of our brain (Fig. 3.1). He assumes low-level processes use the forms of connectionist networks. However, this representation become an obstacle to using high-level ways to think. Consequently, symbolic representations duplicate at larger scales. In the second model, various mental representations are connected with one concept, e.g., `elk` [132]. These representations serve different purposes and together weave a knowledge graph or an encyclopedia. Figure 3.2 is a fragment of the encyclopedia and how-to manual we keep in our heads. When retrieving knowledge about elk, we work at higher levels and would not realize how the word "elk" is actually printed or pronounced. But when we need to leverage these resources, e.g., for motion control, attention can fast switch between different levels of representations.

This perspective of language analysis is so prevalent that many theories of language follow the same pattern: from microscopic to macroscopic. Our intuition from the hierarchical representations of language is that we can extract information from every layer of the structure. The final formation may have fairly big redundancy, while that is how our mind works and how language is used.

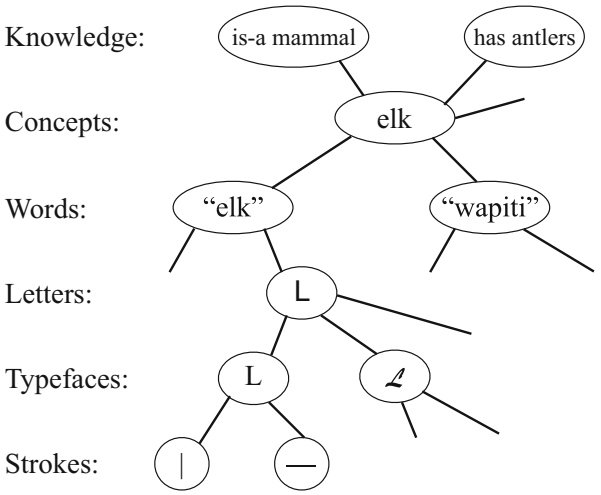

Fig. 3.2 Hierarchical mental representations of concepts [132]

3.2 Three Ways of Looking at the Structure of Language

3.2.1 Lexicon, Grammar, and Pragmatics

The idea that language is a set of lexicons and, at the same time, a syntactic system has been proposed even before the inception of NLP. Aligned with this tradition, the early popular approaches of NLP research as well take a view that emphasizes either the expressiveness [159] or language rules [37]. Lexicography starts by analyzing meaning and formation of words. Word is defined as the smallest element that can be uttered independently with objective or practical meaning, and lexicon is a collection of words. If we investigate the language acquisition process of very young children, the primary expressions they learned is “holophrastic.” People say: “I can’t think of the right word,” but never say “I can’t think of the right prefix” [17].

Word is where most linguists start with semantic analysis. A figurative description of how phrases and sentences are structured is building a house with blocks, while the construction follows certain rules, which are called “grammar.” For example, an adjective can precede a noun, but an adverb cannot precede an adverb. Another example is that a sentence usually has a subject. These syntactic features also contribute to meanings, that is, two sentences can have the same BOW representation but different word sequences. Chomsky [37] believes that these rules are more or less rooted as innate ability of human beings and play an important role for sentence formation.

Part-of-speech (POS) tags assist identifying grammatical features, so do principled types for word combination and dependencies. Figure 3.3 shows various types of grammatical information, including a tree of how a sentence can be decomposed

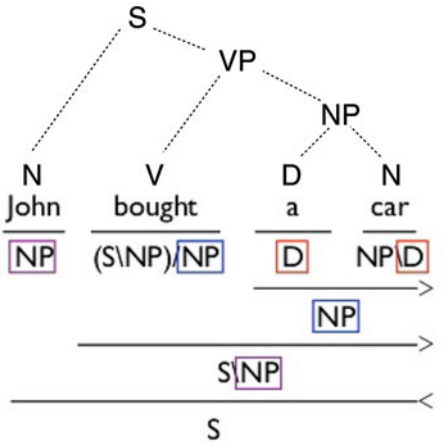

Fig. 3.3 Various types of grammatical information

to a noun and a verb phrase (VP) and a (transitive) verb can take two noun phrase arguments ((S\NP)/NP). Recent studies, e.g. [93], have attempted continuous one-dimensional representations that incorporate this kind of structured information (dependency and constituent grammar).

Beyond grammar are the long-range dependencies in discourses that almost cannot be captured by atomic analysis on sentences. In other words, the context of an utterance provides further information for understanding. This starts to enter the physical world or rhetorics that deviate from the literal meaning, such as implicature, metaphor, and sarcasm [135, 195]. Recent research in dialog systems [20] develops many mechanisms to look up and store key information from conversation history, which is the state-of-the-art model so far to capture long-range dependencies in discourses.

3.2.2 Knowledge, Reasoning, and Emotion

As described in Sect. 3.1, the third layer of the psychological pyramid concerns language, which is the vent of various mental activities. Our mind possesses a huge semantic network representing knowledge types such as "is-a," "has," and other predicates by relations. Knowledge discovery from natural language is the task of identifying resources and extracting these tuples. Automated reasoning based on knowledge unions can exponentially increase the number of knowledge pieces. The following example illustrates how reasoning is conducted:

$$\text{Men are mortal.} \cup \text{Socrates is a man.}$$
$$\Rightarrow \text{Socrates is mortal.}$$

Our mind adapts this process probably because we do not have enough memory to keep records of every knowledge tuple. It is estimated that human beings store around 250,000 knowledge tuples, which includes those important and frequently recalled ones [69]. Other knowledge tuples are derived from reasoning on the existing ones.

Emotions and affective expressions arise only when language is in use, that is, the *ego* is present. Liu [98] echoes this point by defining an opinion as a quadruple (g, s, h, t), where h is the opinion holder and g, s, and t are the sentiment target, sentiment value, and time of expression, respectively; similarly an emotion is defined as a quintuple (e, a, m, f, t), where f is the feeler of the emotion and e, a, m, and t are the target entity, target aspect of the entity, the opinion type, and time of expression, respectively. When negative emotion is conveyed in the sentence "The movie is boring," the latent meaning is that "I think the movie is boring," where "I" is the opinion holder or the feeler of the emotion. Consider an example given by [136]:

"I disliked the movie you love."

The conveyed emotion is negative because "I" is the subject of "disklike." "You" is the subject of "love" but this is less important in the scenario where this conversation happens. We can conclude here that emotion aligns with the pragmatics of language, which is the most advanced layer. This layer is of course supported by words that activate emotional reflection, but cannot be thoroughly explained by low-level features of language.

3.2.3 Yet Another Time Arrow

Our language comprises a large number of time expressions. However, in the abovementioned ways of looking at language, time factor has not been emphasized. Causal-temporal relation is everywhere in our daily life, while little progress has been made on this topic. The definition of emotion includes the time when emotion is expressed because, in examples, time markers like past tense changes the sentiment polarity of emotion [98]. Only until recently, annotation systems for time expressions and time tags have been studied [202].

This time arrow is especially important in financial text analysis because investment opportunities are ephemeral. News impact decays along this arrow and timing of news influences the decision made by investors. Sentiment on the financial market is changing its orientation and intensity for every moment, such that if we relate sentiment to the time arrow, other stable financial information can all be described as semantics. This dichotomy of semantics and sentiment provides the roughest landscape of what can be extracted and analyzed from natural language. In fact, even in the world of finance, it is not difficult to conceive a kind of invariant connection between financial assets, and the separated volatility attributes

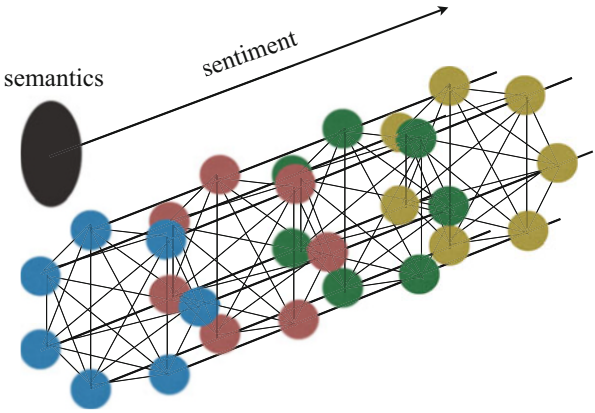

Fig. 3.4 A narrative space for financial information

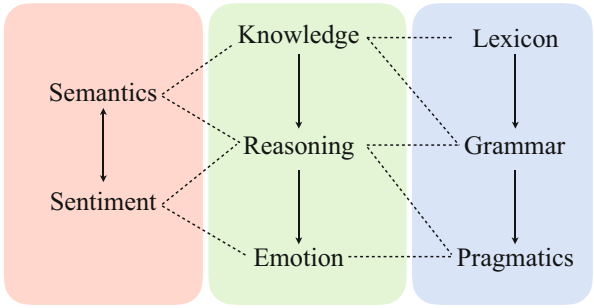

Fig. 3.5 Mapping hierarchical structures of language

to the sentiment (temporal) component (Fig. 3.4). This idea of separating temporal effects from invariance is analogous to CAPM in a sense that expected return is decomposed to a risk-free market return and a risk premium. Outside this narrative space, there is nothing to say.

We make an effort to map among the three ways of describing the structure of language (Fig. 3.5). In the red column, only two dimensions, namely, semantic and sentiment as in Fig. 3.4 are considered. In the green column, we consider the lexical and commonsense knowledge as related to semantics, while sentence level reasoning and discourse emotions are related to financial sentiment. The hierarchy from basic units to high level language structure can also describe the blue column. Notice that this rough mapping is from a perspective of developmental stages and in terms of different scales of analysis. The elements do not have superordinate relations. Particularly, we will extend the semantics-sentiment structure later and show examples of how this mental structure has already been applied to analyze stable and dynamic aspects of financial markets [102].

3.3 Anchor in a Tumultuous Market

Semantic knowledge is recognized as the anchor in a financial market because it serves the basis of our mental activities. Business decisions are made on agglomerating relevant knowledge and planning future actions. Knowledge such as answers to "what is the mission of a company" and "which category does their main business belong to" keeps the same for years and is the starting point for financial statements analysis and other investment behaviors.

To involve computational methods in these activities, machine-readable knowledge representation is a prerequisite. We noticed that there are already business applications of markup language, e.g., Business Intelligence Markup Language (Biml) and eXtensible Business Reporting Language (XBRL). Researchers [161, 197] also tried to enhance the interoperability and build ontologies from the business descriptions. Figure 3.6 provides an example of XBRL snippet describing some accounting information.[1]

However, resistance to adopting these formats exists due to the high cost and fear of losing informational barriers in commercial competitions. Consequently, BOW, TF-IDF, and sub-symbolic methods, which are less resource-dependent against this uncooperative attitude, dominate knowledge representation practice, analysis of social media, and even financial statements.

```
<ifrs-gp:AssetsHeldSale contextRef="Current_AsOf" unitRef="
U-Euros" decimals="0">100000</ifrs-gp:AssetsHeldSale>
<ifrs-gp:ConstructionProCurrent contextRef="Current_AsOf"
unitRef="U-Euros" decimals="0">100000</ifrs-gp:
ConstructionProCurrent>
<ifrs-gp:Inventories contextRef="Current_AsOf" unitRef="
U-Euros" decimals="0">400000</ifrs-gp:Inventories>
<ifrs-gp:OtherFinancialCurrent contextRef="Current_AsOf"
unitRef="U-Euros" decimals="0">50000</ifrs-gp:
OtherFinancialCurrent>
<ifrs-gp:CurrentTaxReceivables contextRef="Current_AsOf"
unitRef="U-Euros" decimals="0">10000</ifrs-gp:
CurrentTaxReceivables>
<ifrs-gp:PrepaymentsCurrent contextRef="Current_AsOf"
unitRef="U-Euros" decimals="0">100000</ifrs-gp:
PrepaymentsCurrent>
<ifrs-gp:CashCashEquivalents contextRef="Current_AsOf"
unitRef="U-Euros" decimals="0">50000</ifrs-gp:
CashCashEquivalents>
<ifrs-gp:AssetsCurrentTotal contextRef="Current_AsOf"
unitRef="U-Euros" decimals="0">810000</ifrs-gp:
AssetsCurrentTotal>
```

Current ASSETS

Assets held for Sale	100,000
Construction in Progress, Current	100,000
Inventories	400,000
Other Financial Assets, Current	50,000
Current Tax Receivables	10,000
Prepayments, Current	100,000
Cash and Cash Equivalents	50,000
Current Assets Total	810,000

Fig. 3.6 An example of XBRL

[1] http://www.investinganswers.com/financial-dictionary/businesses-corporations/xbrl-5714

3.4 Time Series of Asset Return and Sentiment

The application spurt in monitoring public sentiment over the past decade has manifested people's interest in employing this aspect for business and political purposes. Evaluating the effectiveness of financial sentiment is much easier than of semantic representation because temporal effects are added in as basic elements. Imagine the sentiment extracted from financial information as $\mathbb{S}$ and the time series of asset return as $\mathbf{R}$, it seems reasonable to believe that the effectiveness of a representation of $\mathbb{S}$ is justified as we find "$\mathbb{S}$ predicts $\mathbf{R}$." In terms of predictability, we investigate the explanatory power of sentiment and measure of fit of the constructed model.

3.4.1 Predictability: Test of Causality and Residuals

Assume X_t and Y_t are the differenced stationary time series from $\mathbb{S}$ and $\mathbf{R}$; Granger proposed considering two regression models [66]:

$$Y_t = \phi_0 + \sum_{i=1}^{m} \phi_i Y_{t-i} + \epsilon_t \tag{3.1}$$

$$Y_t = \phi_0 + \sum_{i=1}^{m} \phi_i Y_{t-i} + \sum_{j=g}^{h} \psi_j X_{t-j} + \epsilon_t \tag{3.2}$$

Recall the definition of RSS:

$$\text{RSS} = \sum_{t=1}^{N} (Y_t - \hat{Y}_t)^2,$$

The F-statistic, $F = \frac{(RSS_1 - RSS_2)(N-m+h-g)}{RSS_2(h-g)}$, is calculated to test the null hypothesis that $\psi_j = 0, \forall j$. If null hypothesis is rejected, we say X Granger causes Y. The same test is sometimes conducted for Y on X. Consistent with our intuition, it is often reported that Granger causality exists from both sides [19, 157]. This implies the effect of sentiment on asset returns may not be one-directional. However, passing the Granger test is not a strong evidence. Financial time series are almost persistent series and that is when spurious positive results are likely to occur.

It is worth clarifying that the Granger test has some weak points in the sense that it does not help in identification of metaphysical causality and is very strict in a sense that it only concerns first-order statistical causality. It is possible that X provides other information on the distribution of Y through probabilistic reasoning and this still helps in the asset allocation models, but not the price prediction models.

The F-statistic provides some information on the significance of having sentiment in the model. Furthermore, we may want to know if there are other factors and to what extent is the model complete. This is possible by considering the measure of fit. We take ϵ_t from both regressions (3.1) and (3.2) and calculate the $\tilde{\mathrm{Q}}$-statistic, respectively, given by [101]:

$$\tilde{\mathrm{Q}}(residual) = N(N+2)\sum_{k=1}^{m}\frac{residual_k^2}{N-k} \tag{3.3}$$

where $residual_k$ measures the k-th order autocorrelation in residuals

$$residual_k = \frac{\sum_{t=k+1}^{N}\epsilon_t\epsilon_{t-k}}{\sum_{t=1}^{N}\epsilon_t^2}.$$

We can have overall effect of autocorrelations in the residuals by summing them up. Then, the quotient defined as $\tilde{\mathrm{Q}}_1/\tilde{\mathrm{Q}}_2$ can be used as a measure of reduction in autocorrelations of residuals of having X. The larger the quotient is, the more the (non-white noise) predictable component is explained and eliminated by X. Besides the Granger test and measure of fit, the predictability test can be plotted and conducted by rule-of-thumb observations in practice, especially when the information source $\mathbb{S}$ is of high quality.

Chapter 4
Computational Semantics for Asset Correlations

We use a machine, or the drawing of a machine, to symbolize a particular action of the machine.

— Ludwig Wittgenstein

Abstract This chapter explores the possibility to leverage semantic knowledge for robust estimation of correlations among financial assets. A graphical model for high-dimensional stochastic dependence termed a "vine" structure, which is derived from copula theory, is introduced here. To model the prior semantic knowledge, we use a neural network-based language model to generate distributed semantic representations for financial documents. The semantic representations are used for computing similarities between the assets they respectively refer. The constructed dependence structure is experimented with real-world data. Results suggest that our semantic vine construction-based method is superior to the state-of-the-art covariance matrix estimation method, which is based on an arbitrary vine that at least guarantees robustness of the estimated covariance matrix. The effectiveness of using semantic vines for robust correlation estimation for Markowitz's asset allocation model on a large scale of assets (up to 50 stocks) is also showed and discussed.

Keywords Asset allocation · Dependence modeling · Robust estimation · Doc2vec · Semantic vine · Correlation matrix · Machine learning

4.1 Distributed Document Representation

The core idea we propose to measure, the correlation between two financial assets, is based on the concept of semantic linkage. We assume highly correlated assets to be discussed in similar contexts and thus have a strong semantic linkage. Even though various sources of text can be used to together represent financial assets, we consider a more generalized task of first representing gathered texts with a vector. With this transformation, we convert the problem of measuring semantic linkage to a well-studied mathematical problem of similarity measures for real-

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4_4

valued vectors. Although there is a huge amount of information relevant to specific financial assets, we hope to grasp the essence of the information. Li et al. [97] showed a method to summarize the information by a sentence relevance measure, and this summarization works better in his method for stock prediction. Bai et al. [7] developed a business taxonomy that can be strategically used to tag small and innovative companies which do not fit mainstream industry classification systems.

Most of the early studies have employed BOW or bag-of-phrases representations of textual financial data to obtain numerical features, which are more suitable to be processed by computers. These classic techniques already make it possible to agglomerate and analyze a large number of financial articles. The well-known BOW model has been applied to NLP and information retrieval (IR) tasks with a long history. The model represents a piece of text with count statistics, for example, word occurrence frequencies. Noticing that function words such as "a," "the," etc. do not provide useful semantic information, stopword lists are maintained and used to filter out such words. Several drawbacks are realized when using the BOW model. One of them is that natural languages have very large vocabularies that keep growing with global communication. As a result the vector representations of sentences and texts are sparse. Without a level of understanding that goes beyond symbols and strings, many texts do not show any similarity. Another disadvantage is that the word order information is also not taken into consideration in the BOW model. This in certain cases causes problem. For example, the financial news "Samsung now is gaining advantages on Apple" and "Apple now is gaining advantages on Samsung" would lead to opposite market reactions, though they share the same bag-of-words representation. This is because of different grammatical roles and POS taken by the same word. In the above examples, "Samsung" is the subject in the first sentence while a noun modifier in the second. A way to preserve the word order and contextual information is to use bag-of-n-grams instead of bag-of-words, but with bag-of-n-grams the dimension of the vector representation exponentially explodes. The semantic gap between different expression entries as mentioned before is also a common phenomenon observed from financial texts. For instance, when similar gists are phrased by different words, such as in two news titles "Brexit caused a drop in the pound" and "Leaving the EU accelerates pound's slump," this semantic similarity cannot be captured; another related research gives an example of words like *strong* and *powerful*, which are counted as different dimensions [89]. Whereas in reality, there should be a link between the semantics they carried.

These problems with classic text representation techniques require a denser, fixed-length, continuous real-valued representation of words and texts that well addresses the importance of contexts. With the advance of machine learning, especially deep learning, this representation can be obtained in multiple ways, for example, during the training process of a neural language model (*skip-gram word2vec*) [12, 114] or autoencoder. The former begins with a sparse matrix where each row or column is the index of a word, hence has a dimension of the vocabulary

size. A shallow neural network[1] is trained with the learning objective of maximizing the average word occurrence probability given its context window:

$$\max_{U,b} \frac{1}{T} \sum_{t=k}^{T-k} \log \Pr(x_t | x_{t-k}, \ldots, x_{t+k}) \tag{4.1}$$

where U and b are neural network parameters to be trained; $\{x_1, x_2, \ldots, x_T\}$ is the word list from training corpus; k is the (single-side) context window length. Parameters U and b are iteratively adjusted via back-propagation of word prediction errors [12, 89], and the gradients are produced by optimizers. Stochastic gradient descent (SGD) is the most commonly used optimizer, while when faster convergence is favored, other optimizers such as Adam [85] or AdaBound [104] are used as well. With the probabilistic settings, we define the prediction error as a difference of the estimated and the true log probabilities. Then the update rule is the following:

$$\Delta(U, b) = -\epsilon \frac{\partial \log \hat{\Pr}(x_t | x_{t-k}, \ldots, x_{t+k})}{\partial (U, b)} \tag{4.2}$$

When the training process is finished, the neuron weights form a space that distributionally represents words' contexts. Another interpretation is that the parameters together represent the word with a fixed-length vector which has the same size as the neural network layer. This interpretation resonates with many structuralism theories of lexical semantics, such as [40, 91]. Empirical analysis also supports the claim that this method of word representation well-conserves semantic relations.

Figure 4.1 adapted from Mikolov et al. [114] illustrates the analogous ability of word embedding. That is, projecting the symbolic representations of words to a computationally viable space. When visualizing the country name and its corresponding capital name pairs, the vectors representing this relationship are almost parallel and of the same length. Extending this word representation idea, a sequel to the word2vec model [89] proposed a distributed representation for a document (doc2vec). The model outperforms simply averaging the word vectors that appear in the document. Its idea is simple: to create "new words" termed "tokens" associated with documents. Then, a similar neural language model is trained by learning parameters that simultaneously represent a document and its token. Referring to equation 4.1, this means that for a certain word x_t belonging to document D_i, its context will no longer be $(x_{t-k}, \ldots, x_{t+k})$, but $(x_{t-k}, \ldots, x_{t+k}, d_i)$ instead. The additional context d_i is the token. With this change of equation 4.1, the learned representations of tokens well reflect the topic of its associated document. The entire document vector hence has a fancy name called *distributed memory*. With this model, we consider a new document that does not appear in the training set. After

[1] Shallow refers to the neural networks that only have one hidden layer of neurons.

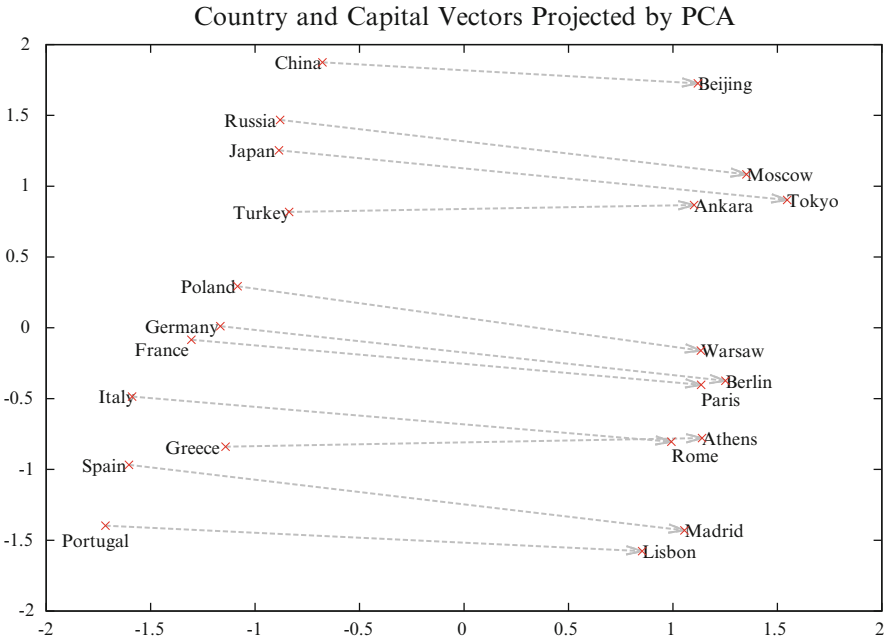

Fig. 4.1 Ability of analogously organizing concepts and learn relationships by word embedding [114]

concatenating all the word vectors, we can add one virtual token to represent the semantic distribution at a document level, so the new document can be represented by others.

4.1.1 Similarity Measure for Assets

The document-embedding technique (doc2vec) enables representing documents with different lengths. Therefore, we have the motivation to include as much useful information and build an overall descriptive document from many relevant documents pinpointed to each asset a_i and compute a vector representation $\boldsymbol{vector}(a_i)$ that preserves the semantics. Subsequently, the semantic linkage between two assets will align with the vector similarity of their descriptive document representations and can be used for asset dependence modeling.

We use the cosine similarity to estimate pairwise semantic linkage for asset pair a_i and a_j:

$$\begin{aligned} s(a_i, a_j) &= \cos < \boldsymbol{vector}(a_i), \boldsymbol{vector}(a_j) > \\ &= \frac{\boldsymbol{vector}(a_i) \cdot \boldsymbol{vector}(a_j)}{||\boldsymbol{vector}(a_i)||_2 ||\boldsymbol{vector}(a_j)||_2}. \end{aligned} \tag{4.3}$$

In the rest of this chapter, we denote pairwise semantic linkage $s(a_i, a_j)$ with a short-form s_{ij}. Noticing that the vector representations for assets are produced by softmax functions, each dimension has a value between 0 and 1; therefore s_{ij} is also between 0 and 1.

4.2 Vine Dependence Modeling

Recall Markowitz's model of Sect. 2.3.1: the framework which maximizes the portfolio return and minimizes the portfolio risk. In the model asset, expected returns ($\boldsymbol{\mu}$) and correlations ($\boldsymbol{\Sigma}$) of assets are important inputs and need to be accurately estimated because the average expected returns reflect the profitability of the portfolio, while the overall portfolio risk depends on the correlations between pairs of assets. However, both these variables are difficult to estimate accurately from historical return time series in practice because discrete observed asset returns are peculiarly distributed. Figure 4.2 showcases the typical skewed and fat-tailed distribution of realized daily return ratio of Apple Inc's stock price. Apparently, it cannot be approximated with a Gaussian distribution as suggested by the Markowitz's model, and the same difficulty exists for identification and approximation of the expected return distribution of most of the stocks. Empirical study of US stocks [139] favors the stable Paretian hypothesis over Gaussian hypothesis. Student's t-distribution on low degrees of freedom or Cauchy distribution are commonly considered as well, where mean and variance are undefined. In fact, the standard deviation (std) of returns in Fig. 4.2 (red) is around four times that of the Gaussian distribution fitting depicted by the blue curve. Unfortunately, the situation

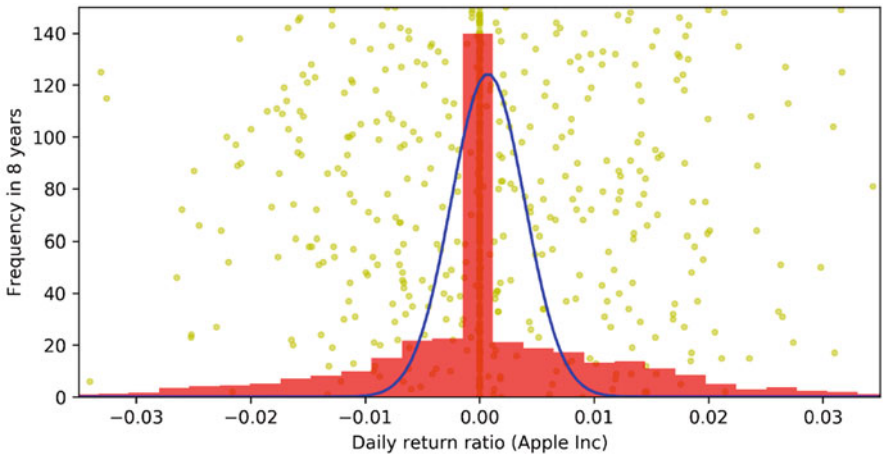

Fig. 4.2 Real distribution and Gaussian fitting of returns of Apple's stock price (2009–2017) [191]

is becoming increasingly challenging today in an interconnected world. According to the Schwab Center for Financial Research [43], the average correlation between four US equity classes increased from around 0.65 to 0.90 in the last 20 years. These facts motivate us to study robust estimation of asset correlations.

Robust estimation of the correlation matrix is not easy, especially in high-dimensional cases. This is because in multivariate analysis, one has to not only consider the estimated variable itself but also its connection to other variables. For correlation matrix estimation, the majority of the methods are based on combining pairwise correlation estimations. However, by simply using robust pairwise correlation estimates as matrix elements, positive definiteness of the estimated correlation matrix cannot be guaranteed. Because the covariance matrix is positive-definite by definition, violating this property will lead to difficulty and undesired computing outcomes, e.g., the risk measurement according to the modern portfolio theory may become negative. Negative risk means that the optimized portfolio weights will be a corner solution of the feasible domain, that is, the benefits from diversification would disappear. Even a positive-definite estimation requires robustness; otherwise using the unstable estimation of asset correlation matrix will cause the optimized weights to be extremely large, and frequent rebalancing of the portfolio will be needed for multi-period applications.

Previous studies have migrated many classic techniques in robust statistics for the estimation of asset expected returns and their covariance matrix. Such techniques include trimming, quantile statistics, M-estimators, minimum covariance determinant (MCD), minimum volume ellipsoid (MVE), iterated bivariate Winsorization (IBW), and more [121, 138, 183]. There are two schools that hold opposite opinions on the importance and difficulty of the two related tasks [191]. One argues that expected asset returns are more important in the Markowitz's model and more difficult to be accurately estimated since numerous exogenous variables can have effect on the expected returns [112, 138]; however, the other (mainly statisticians) argues that the covariance matrix is a more challenging object to estimate because it involves a quadratic number of parameters ($\mathcal{O}(n^2)$) from the dependence structure of the assets [204]. We attempt to tackle both problems and leave the return estimation problem to the next chapter. Subsequently, we focus on estimating a robust covariance matrix for asset returns in the rest of this chapter.

The covariance matrix estimation task does differ from the expected return estimation under the perspective of narrative space for financial information (see Fig. 3.4). The sentiment-driven price movements feature fast adaptation to the financial news and volatile changes along the time arrow. In comparison, the dependence relations between assets are more stable. They tend to be affected by macroeconomic and intrinsic factors, such as what industry they belong to, their products, and their position in a supply chain. This type of information stays unchanged for years. The knowledge that expected asset returns and the covariance matrix require different information to be estimated is not taken into consideration by Markowitz's framework of analysis. The CAPM model [171], which tries to connect equilibrium return and risk, also fails to use such knowledge. In our model, we thus leverage the semantic prior knowledge as a solution to covariance

matrix estimation. To ensure the robustness of this process, we first solve a vine selection problem. The formed vine structure is later used to induce positive-definite estimation of the covariance matrix.

4.2.1 Copula and Vine Decomposition

We think of each financial asset's expected return as a random variable μ_i and their joint distribution $p(\mu_1, \mu_2, \ldots, \mu_n)$. Because of the dependence between asset returns, the joint distribution cannot be directly factorized to the marginal distribution of each asset's return. The function that links the joint distribution and its marginal distributions is called a copula density, formally,

$$p(\mu_1, \mu_2, \ldots, \mu_n) = [\prod_{i=1}^{n} p(\mu_i)] \; c(\mu_1, \mu_2, \ldots, \mu_n). \tag{4.4}$$

Describing and learning high-dimensional copulas are difficult. For example, visualizing such copulas needs a lot of computing power; measuring the difference between two high-dimensional copulas by Kullback-Leibler distance also involves calculation of a multiple integral, which is not tractable. Therefore, many techniques are developed to decompose high-dimensional copulas, such as copula trees, vine copula, and copula Bayesian networks. We study here the vine decomposition of copulas, which manages to represent a high-dimensional copula by multiple bivariate copulas as basic building blocks. Specifically, Archimedean bivariate copula families, such as the Clayton copula, the Gumbel copula, and the Frank copula, are most widely applied as they can be parameterized by a single value [51]:

$$\begin{aligned}
c^{Clayton}(\mu_1, \mu_2|\theta) &= (\mu_1^{-\theta} + \mu_2^{-\theta} - 1)^{-1/\theta} \\
c^{Gumbel}(\mu_1, \mu_2|\theta) &= e^{-[(-\log \mu_1)^\theta + (-\log \mu_2)^\theta]^{1/\theta}} \\
c^{Frank}(\mu_1, \mu_2|\theta) &= -\theta^{-1} \log(1 + \frac{(e^{-\theta\mu_1} - 1)(e^{-\theta\mu_2} - 1)}{e^{-\theta} - 1}).
\end{aligned} \tag{4.5}$$

We show how bivariate copulas can form higher-dimensional copula via conditioning [10]. Consider copula density $c(\mu_1, \mu_2, \mu_3)$, if we define $c_{13|2} := c(\mu_1, \mu_3|\mu_2)$ as $\frac{c(\mu_1,\mu_2,\mu_3)}{c(\mu_1,\mu_2)c(\mu_2,\mu_3)}$, we see that $c_{13|2}$ is also a bivariate copula of variables $\mu_1|\mu_2$ and $\mu_3|\mu_2$. Therefore, we have $c_{123} = c_{12}c_{32}c_{13|2}$. Obviously, by strategically choosing the conditioning variables, which also implicitly defines a vine structure, one can decompose a higher-dimensional copula to the product of multiple bivariate copulas:

$$c(\mu_1, \mu_2, \ldots, \mu_n) = \prod_{i=1}^{n-1} \prod_{j,\, k} c(j, k|D(e)), \tag{4.6}$$

where $D(e)$ is the conditioning set.

4.2.2 Vine Structure and Its Properties

The interactive pattern among multiple variables grows exponentially complicated as the dimension increases. To describe the joint distribution representation, the mathematics of high-dimensional dependence modeling extends bivariate dependence measures with graphical models. Such models include Bayesian belief networks [33] and Markov random fields [113] for causal effects and vine structures for stochastic dependence. Unlike common graphical models, a vine is unique for its recursive tree-like structure, where the edges of nodes become nodes for the next tree.

As a result, a vine can describe more types of high-dimensional dependency patterns by combining basic elements compared to most of the distribution families. For this property, vine structures are frequently applied in statistical machine learning and financial modeling [52, 170]. In theory, a good vine structure should be of a proper level of complexity. That is, it provides not every aspect to determine the exact distribution (over-fitting). However, it provides all the important slices of the high-dimensional joint distribution using as few as possible parameters. Figure 4.3 adapted from [204] provides an example of how the delicate dependency pattern among three financial assets can be depicted.

In Fig. 4.3, the pairwise correlations between a_1 and a_2 and a_2 and a_3 are both 0.8, which means that the returns of the paired two assets are positively correlated. However, the partial correlation between a_1 and a_3 condition on a_2 is -0.8, which means that returns of a_1 and a_3 or negatively correlated.[2] This case is possible when a_2 represents a company selling costumer products, e.g., Apple, while a_1 and a_3 are competitive suppliers to a_2, e.g., TSMC and Foxconn [204]. In this case a_1 and a_3 are called first-order tree structure and a_2 describes the second-order dependence.

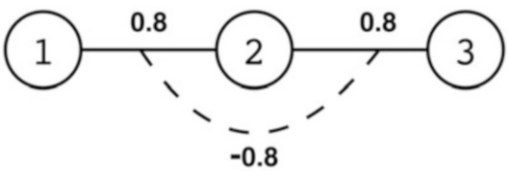

Fig. 4.3 Example of a vine structure on three financial assets

[2]The correlation between a_1 and a_3 conditions on a pivot asset a_2. Therefore, we use a different type of dashed link to denote this conditional correlation. The dashed link is abbreviated as 1, 3|2.

A vine structure degenerates to a (first-order) Markov tree when only the first-order tree structure is specified. In a Markov tree, long-range dependence does not exist. In the following when we discuss vine structures, we refer to fully linked vines with every order of tree structure specified by default. While the fundamental definitions and theorems can be found in relevant books [87], our contributions are mainly on the proposal of Theorem 4.3 and Definition 4.5.

First, we recall the following definition of vine by Bedford and Cooke [11]:

Definition 4.1 A vine $\mathscr{V} = \{T_1, T_2, \ldots, T_n\}$ is a set of linked trees on n elements if:

- T_1 is a tree with nodes $N_1 = \{1, 2, \ldots, n\}$ and a set of edges denoted by $\mathscr{E}_1$.
- For $i = 2, \ldots, n-1$, T_i is a tree with nodes $N_i = \mathscr{E}_{i-1}$ and edge set $\mathscr{E}_i$.

Specially, a vine is called a *"regular vine"* if for any tree T_j, the nodes of an edge in edge set $\mathscr{E}_j$ share one and only one node in common. This rule is called the *"proximity condition of regular vines."* A regular vine has some good properties similar to what a binary tree has for data structure. Mathematically, the regular vine can guarantee the interpretation of edge weights as partial correlation coefficients, conditioning on the shared spanning variables. We go on to investigate two special types of regular vine (aka C-vine and D-vine) defined by Aas and Berg [1]:

Definition 4.2 A regular vine is called a **Canonical** or **C-vine** if each tree T_i has a unique node of degree $n - i$ for $i = 1, \ldots, n-2$, where the unique node is called the **root** for each tree.

Definition 4.3 A regular vine is called a **Drawable** or **D-vine** if each node in T_i has a degree of at most 2, for $i = 1, \ldots, n-1$.

Figure 4.4 gives examples of C-vine and D-vine on four nodes. Notably, if a regular vine only has three nodes in T_1, it is simultaneously both a C-vine and a D-vine. The copula decompositions based on the C-vine and the D-vine of Fig. 4.4 are thus:

$$\begin{aligned} c_{1234}^{C-vine} &= c_{24|13} c_{23|1} c_{34|1} c_{12} c_{13} c_{14} \\ c_{1234}^{D-vine} &= c_{14|23} c_{13|2} c_{24|3} c_{12} c_{23} c_{34}. \end{aligned} \tag{4.7}$$

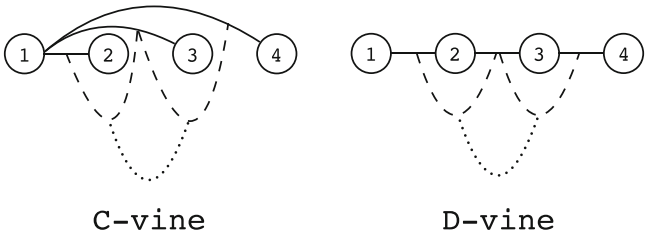

Fig. 4.4 Examples of C-vine and D-vine

Although the bivariate copulas still need a heavy amount of information to specify, we simplify the specification using only partial correlations of the two variables. Therefore, each edge can be specified by a single parameter, namely, the partial correlations. We introduce the formal definition of a partial correlation vine as follows [204]:

Definition 4.4 A partial correlation vine is a vine where each edge in the edge set $\mathscr{E}(\mathscr{V})$ is assigned a partial correlation value ρ between -1 and 1.

Finally, we have the following important Theorem 4.1 that guarantees robust correlation matrix estimation on a partial correlation vine according to the above-mentioned definitions.

Theorem 4.1 *For any regular vine on n elements and the set of partial correlation specifications for the vine, there exists a $n \times n$ positive-definite correlation matrix and vice versa.*[3]

For a partial correlation C-vine or D-vine, an analytical solution exists for each element in the full correlation matrix to be computed from just the set of partial correlations according to Bedford and Cooke [11]. However, for general regular vines, the computation has to be step-by-step on subvines as the decomposition is difficult to be expressed in a closed form. The following Theorem 4.2 adapted from Lemma 13 in [11] facilitates this step-by-step computation based on subvines. We restate Theorem 4.2 here without a proof.

Theorem 4.2 *Let Σ be the covariance matrix of n joint normal distributed random variables. Write Σ_A for the principal submatrix built from row 1 and row 2 of Σ, etc. so that*

$$\Sigma = \begin{bmatrix} \Sigma_A & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_B \end{bmatrix}.$$

Then the conditional distribution of elements 1 and 2 is normal and the covariance matrix has the form:

$$\Sigma_{12|3\ldots n} = \Sigma_A - \Sigma_{AB}\Sigma_B^{-1}\Sigma_{BA}. \tag{4.8}$$

Theorem 4.3 is then derived based on Theorem 4.2.

Theorem 4.3 *Consider a subvine of only three nodes 1, 2, and 3, where node 2 is the root. The unconditional correlation of 1 and 3 can be calculated from their correlation conditional on 2 and their partial correlations with 2:*

$$\rho_{13} = \rho_{13|2}\sqrt{(1-\rho_{12}^2)(1-\rho_{23}^2)} + \rho_{12}\rho_{23}. \tag{4.9}$$

[3] Proof of this theorem uses trigonometric substitution. For details, see Lemma 12 in Bedford and Cooke [11].

Proof We write the partial correlation matrix as

$$\boldsymbol{\rho} = \begin{bmatrix} 1 & \rho_{13} & \rho_{12} \\ \cdot & 1 & \rho_{23} \\ \cdot & \cdot & 1 \end{bmatrix}.$$

We apply equation 4.8 with $\Sigma_B = [1]$ and

$$\Sigma_A = \begin{bmatrix} 1 & \rho_{13} \\ \rho_{13} & 1 \end{bmatrix}, \qquad \Sigma_{AB} = \begin{bmatrix} \rho_{12} \\ \rho_{23} \end{bmatrix}, \Sigma_{13|2} = \begin{bmatrix} \sigma_{1|2}^2 & \rho_{13|2}\sigma_{1|2}\sigma_{3|2} \\ \rho_{13|2}\sigma_{1|2}\sigma_{3|2} & \sigma_{3|2}^2 \end{bmatrix},$$

then we will have:

$$\begin{bmatrix} \sigma_{1|2}^2 & \rho_{13|2}\sigma_{1|2}\sigma_{3|2} \\ \rho_{13|2}\sigma_{1|2}\sigma_{3|2} & \sigma_{3|2}^2 \end{bmatrix} = \begin{bmatrix} 1 & \rho_{13} \\ \rho_{13} & 1 \end{bmatrix} - \begin{bmatrix} \rho_{12} \\ \rho_{23} \end{bmatrix} \begin{bmatrix} \rho_{12} & \rho_{23} \end{bmatrix},$$

where we derive the following equations:

$$\sigma_{1|2}^2 = 1 - \rho_{12}^2 \tag{4.10}$$

$$\sigma_{3|2}^2 = 1 - \rho_{32}^2 \tag{4.11}$$

$$\rho_{13} = \rho_{13|2}\sigma_{1|2}\sigma_{3|2} + \rho_{12}\rho_{23}. \tag{4.12}$$

Substituting $\sigma_{1|2}$ and $\sigma_{3|2}$ in equation 4.12 with equations 4.10 and 4.11, we can get equation 4.9. □

4.2.3 Growing the Semantic Vine

Previous applicational studies frequently encounter the challenge of determining an appropriate vine structure because only after that bivariate copula can be estimated. This task is nontrivial as one "generator" vine structure corresponds to an infinite number of joint distributions. One popular solution is to resort to domain experts and the other solution being to simply assume a C-vine or D-vine structure for the simplicity of computation (because C-vine and D-vine are structures where analytical decomposition is possible) and then discuss the properties (such as associated correlations or bivariate copulas) of edges. However, if we attempt to find the most appropriate vine structure, for n elements there exist $4n!\sqrt{2^{n(n-5)}}$ regular vines [39], only a few out of which are tailored to best describe the high-dimensional probability distribution. To identify such a vine structure, Kurowicka and Joe [87] proposed a *top-down* approach for regular (partial correlation) vine growing. This approach splits nodes into two groups or subgroups for each layer and

ensures that the absolute values of the partial correlations between the "splitting" nodes are the smallest.[4] However, choosing such "splitting" nodes is only possible when the entire correlation matrix is fully specified. Since we are unable to robustly estimate the partial correlation matrix, and our purpose of having this vine structure is exactly the same; the fact that the resulting vine structure might not be robust as well causes a circular explanation. As a reply to this problem, our construction of the vine-growing task is a *bottom-up* method. We propose to grow a semantic vine by first building edges between individual assets with a strong semantic linkage, so that specification of partial correlations with the largest absolute value are prioritized [191]. Algorithm 4.1 elaborates the iterative execution of this process for each layer.

Algorithm 4.1 first ranks the node pairs according to their semantic linkage and checks the adjacency of two nodes every time before growing an edge. If the candidate edge is illegal to be added, the algorithm considers the next pairs. As a result, the algorithm ensures that the semantic vine is a regular vine, but it is not necessarily a C-vine or D-vine. For each layer, the termination condition checks the degree of every node. This ensures that the semantic vine will be fully connected with no isolated node.

The formal definition of a semantic vine is given as follows:

Definition 4.5 A semantic vine is a regular partial correlation vine where the partial correlation values of edges are estimated from pairwise semantic linkages.

Definition 4.5 gives the following properties to our semantic vine. Each edge in the semantic vine is associated with a value ϖ_{ij} that represents the conditional semantic partial correlation and is calculated from the semantic linkage s_{ij} between the two assets that the edge is spanning:

$$\varpi_{ij} = (2 * s_{ij} - 1)/(1 + \varepsilon). \tag{4.13}$$

Since the semantic linkage s_{ij} is in the range of $[0, 1]$ and ε is a small positive scaling factor, equation 4.13 maps the semantic linkage to a financial linkage that is strictly in a range of $(-1, 1)$. In real-life cases, semantic-irrelevant assets are often regarded as safe-haven choices from different industries. Therefore, their asset returns would demonstrate reverse co-movements [191].

4.2.4 Estimating the Robust Correlation Matrix

In this section we first study the inverse process of robust correlation matrix estimation, that is, given a prior correlation matrix, we attempt to find an appropriate

[4]This is defined as the optimal truncation of vines as a minimum number of edges would have large absolute partial correlations and rest of the edges are assumed insignificant (independent).

Algorithm 4.1: Growing semantic vine structure

Data: asset list $\boldsymbol{a}$, pairwise semantic linkage s_{ij}, $i, j = 1, 2, \ldots, n$
Result: semantic vine $\mathscr{V}_s = \{T_i, N_i, \mathscr{E}(\mathscr{V}_s)\}$

```
for k = 1, 2, ..., n − 1 do
    list(s_ij) ← descending sort s_ij(i < j);
    if k > 1 then
        N_k = E_{k−1};
    end
    repeat
        re-compute list(s_ij)|N_k;
        if adjacent(i, j) in N_k then
            E_k ← max(list(s_ij));
            if ∃ loop in E_k then
                discard max(list(s_ij)) from E_k;
            end
            delete max(list(s_ij)) from list(s_ij);
        end
    until ∄ degree(N_k) = 0;
end
return V_s;
```

vine structure to model it. This task is called vine truncation and selection [87, 128]. We directly estimate the robust correlation matrix from the semantic vine (as a prior) obtained from Algorithm 4.1.

It is worth noticing that without the intervene of a semantic vine, neither the pairwise semantic linkage matrix S nor the semantic partial correlation matrix $\wp$ can guarantee positive definiteness. Therefore, to write $\boldsymbol{\rho} = \wp^{-1}$ is not robust (or even viable). In fact, we are ignorant of whether the inverse of matrix $\wp$ exists or not. However, Theorem 4.3 guarantees the existence and robustness of the full correlation matrix $\boldsymbol{\rho}(\mathscr{V}_s)$ without specifying the estimation procedure. With the help of Theorem 4.3, we demonstrate this procedure of step-by-step element-wise estimation running on subvines in Algorithm 4.2. Because the correlation matrix is symmetric, we only care about the upper triangular part ($i < j$).

The asset pairs (i, u) and (j, v) in Algorithm 4.2 exist and are unique. This is because of the definition of the structure of a regular vine. The nodes in $\mathscr{E}_k$ are inherited from $\mathscr{E}_{k-1}$. Assuming there are (i, u) and (i, u') both in $\mathscr{E}_{k-1}$, $u \neq u'$, then for edge set $\mathscr{E}_k$, we will have $(u, u')|i, \ldots \in \mathscr{E}_k$, instead of $(i, j) \in \mathscr{E}_k$. A similar statement holds for asset j.

Because by definition u and v will satisfy that edge (i, u) and (j, v) are in $\mathscr{E}_{k-1}$, we know that ρ_{iu} and ρ_{jv} have already been computed along with a smaller index k of edge set. Thus the order of computing ρ_{ij} in Algorithm 4.2 actually guarantees that each ρ_{ij} is computable, and there are $n + (n-1) + (n-2) + \ldots = \frac{(n+1)n}{2}$ times of value assignment in total. This covers all the unique values required in the symmetric correlation matrix.

Algorithm 4.2: Estimating robust correlation matrix

Data: semantic partial correlation matrix $\wp$, semantic vine $\mathscr{V}_s$
Result: correlation matrix $\boldsymbol{\rho}_{n\times n}(\mathscr{V}_s)$

for $i = 1, 2, \ldots, n$ **do**
  $\rho_{ii} \leftarrow 1$;
end
for $(i, j) \in \mathscr{E}_1$ *and* $i < j$ **do**
  $\rho_{ij} \leftarrow \varpi_{ij}$;
end
for $k = 2, 3, \ldots, n-1$ **do**
  for $(i, j) \in \mathscr{E}_k$ *and* $i < j$ **do**
    $u \leftarrow (i, u) \in \mathscr{E}_{k-1}$;
    $v \leftarrow (j, v) \in \mathscr{E}_{k-1}$;
    $\rho_{ij} \leftarrow \varpi_{ij}\sqrt{(1-\rho_{iu}^2)(1-\rho_{jv}^2)} + \rho_{iu}\rho_{jv}$;
  end
end
return $\boldsymbol{\rho}(\mathscr{V}_s)$;

4.3 Data Used for Experiments

A list of 55 stocks from the US markets[5] is investigated. Because most of the stocks are of famous and big companies, a large percentage of the list overlaps with the lists of stocks investigated by Zhang [200] and Zhu [204]. The industry classification codes of our list of stocks are manually retrieved from the Bloomberg Terminal and Thomson Reuters Eikon in 2017. Furthermore, we get the historical closing prices of five randomly selected stocks from the list. The data is crawled from Quandl API[6] based on which we construct a virtual portfolio. The tickers and corresponding numbering of the five stocks are: Apple Inc (`1:AAPL`), Microsoft Corporation (`2:MSFT`), Goldman Sachs Group Inc. (`3:GS`), Pfizer Inc. (`4:PFE`), and Wells Fargo & Company (`5:WFC`). The price data is later processed with the Markowitz's mean-variance optimization (MVO) method.

The semantic vector embedding space for financial document representation is trained with language materials from two sources: (1) the public available Reuters-21578 Corpus,[7] which contains 10,788 financial news documents totaling 1.3 million words and (2) the Wikipedia pages of our list of stocks.[8]

We use the Reuters Company Business Descriptions, which are paragraphs of brief summarization of the company's business scope, to generate dense semantic vector representations (100-dimensional) for each stock. To get a glimpse of the

[5] Information on the stock list is elaborated in Appendix A.

[6] http://quandl.com/tools/api

[7] http://daviddlewis.com/resources/testcollections/reuters21578

[8] Retrieved from the Internet on 2017-10-09.

Table 4.1 Keywords used to generate vector representations for the selected stocks [191]

Stock ticker	Keywords
`1:AAPL`	Apple, design, mobile, device, digital, computer, iPhone, software, service, store, application, accessory, support
`2:MSFT`	Microsoft, technology, software, business, productivity, develop, system, manufacture, device, computer, solution, intelligent
`3:GS`	Goldman, Sachs, investment, bank, management, client, institutional, financial, advisory, security, loan, asset, service
`4:PFE`	Pfizer, research, pharmaceutical, healthcare, medicines, vaccines, inflammation, business, generics, consumer, immunology
`5:WFC`	Wells Fargo, wholesale, bank, wealth, financial, service, investment, management, commercial, mortgage, retail

content of such descriptions, we present the automatically extracted keywords for the selected stocks we used to construct the virtual portfolio in Table 4.1. After we compute the vector representations of the descriptive stock company profiles, we can use the cosine similarities to form the pairwise semantic linkage matrix:

$$S = \begin{bmatrix} 1 & 0.7084 & 0.3826 & 0.1992 & 0.4544 \\ \cdot & 1 & 0.4856 & 0.3874 & 0.5620 \\ \cdot & \cdot & 1 & 0.3072 & 0.4910 \\ \cdot & \cdot & \cdot & 1 & 0.3767 \\ \cdot & \cdot & \cdot & \cdot & 1 \end{bmatrix},$$

and thus the partial correlation matrix:

$$\varpi = \begin{bmatrix} 1 & 0.4167 & -0.2267 & -0.6426 & -0.0304 \\ \cdot & 1 & -0.0347 & -0.2253 & 0.1240 \\ \cdot & \cdot & 1 & 0.2909 & -0.0179 \\ \cdot & \cdot & \cdot & 1 & -0.2747 \\ \cdot & \cdot & \cdot & \cdot & 1 \end{bmatrix}.$$

We observe from the semantic linkage matrix S that the strongest semantic linkage is between Apple and Microsoft (0.7084), which are both software manufacturers and belong to the technology sector. While the weakest linkage is between Apple and Pfizer (0.1992), which are in totally different business sectors (software and technology vs. pharmaceutical).

4.4 Experiments

Our semantic vine growing and correlation matrix estimation methods on stock price data are evaluated based on three experiments. The initial experiment tests several portfolio settings using either robust or non-robust correlation matrix estimation without any vine structure. The second experiment compares the portfolio performance of using the semantic vine with using a number of arbitrary vine structures. Finally, the last experiment addresses the scalability of our method and its application to financial knowledge discovery.

4.4.1 Obtaining the Semantic Vine and Asset Correlation Matrix

A step-by-step demonstration is provided on how we obtain the semantic vine structure for the selected stocks (Fig. 4.5) using Algorithm 4.1. Note that each edge in a low-level tree is treated as a node in the higher-level tree.

Apparently, the resulting vine structure for our portfolio is neither a C-vine nor a D-vine. Tree 1 mixes the subvine structure of C-vine and D-vine, that is, nodes {1, 2, 4, 5} resemble a C-vine and nodes {1, 2, 5, 3} or nodes {4, 2, 5, 3} resemble a D-vine; Tree 2 has a C-vine structure; Tree 3 and Tree 4 follow a D-vine structure. The logical structure of C-vine and D-vine actually reflects different aspects of the joint distribution of variables. A C-vine is more appropriate for dependence modeling when a critical variable "*leads*" the others, whereas a D-vine is more suitable when the variables are relatively equal/independent. In our semantic vine here, Microsoft

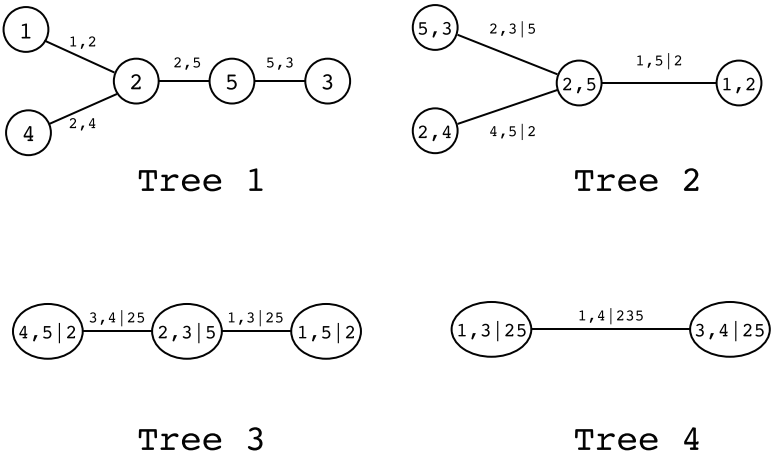

Fig. 4.5 The semantic vine constructed for the stocks [191]

and Wells Fargo are to some extent "*hubs*" that bridge other stocks in Tree 1 and Tree 2. Another interpretation is that the least related nodes are joined in the highest order Tree 4. In our case these nodes are numbers 1 and 4, which represent Apple and Pfizer, which also have the minimum s_{ij} in the semantic matrix. In this sense, our semantic vine-growing algorithm produces a similar structure to the optimal vine truncation suggested by [87] in this specific case, but with more theoretical soundness.[9]

We calculate the unconditional full product-moment correlation matrix as a robust correlation matrix estimator for the stocks using Algorithm 4.2:

$$\boldsymbol{\rho} = \begin{bmatrix} 1 & 0.4167 & -0.2267 & -0.6426 & -0.0304 \\ \cdot & 1 & -0.0347 & -0.2253 & 0.1240 \\ \cdot & \cdot & 1 & 0.2909 & -0.0179 \\ \cdot & \cdot & \cdot & 1 & -0.2747 \\ \cdot & \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

4.4.2 Robust Asset Allocation

Since the positive definiteness of matrix $\boldsymbol{\rho}(\mathscr{V}_s)$ is ensured by Theorem 4.1, we use $\boldsymbol{\rho}(\mathscr{V}_s)$ as the static covariance matrix estimator of asset returns in equation 2.7:

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{\rho}(\mathscr{V}_s). \tag{4.14}$$

This setting can produce meaningful portfolio weights using Markowitz's model. Another key ingredient for Markowitz's model is the expected return. First, daily returns are calculated as:

$$\boldsymbol{R}_d = \frac{\pi_d - \pi_{d-1}}{\pi_{d-1}}, \tag{4.15}$$

then the portfolio expected return is estimated by averaging R_d in a time period of length k:

$$\hat{\mu} = \frac{1}{k} \sum_{d=1}^{k} R_d. \tag{4.16}$$

[9] See Sect. 4.2.3 for the definition of the optimal vine truncation.

Below we compare five models with different estimators (EW, rMVO, MVO, drMVO, dMVO) and with the assumption of no short selling of stocks, taxes, or transaction cost:

1. The equal-weighted portfolio (EW): EW refers to the case that each asset has the same portfolio weight. In our portfolio where five assets are considered, we use a holding weight of [20%, 20%, 20%, 20%, 20%] throughout the test period. Consequently, the portfolio return will be an average of the individual asset returns. EW is a simple yet tough-to-beat baseline. Empirical study [55] shows that the equal-weighted portfolio outperforms many other active asset management strategies.
2. Robust MVO (rMVO): This refers to the mean-variance optimization method with both a robust covariance matrix estimation (based on the semantic vine) and robust return estimations. We use static estimations calculated by averaging returns in the past 30 days from the starting data of test period and use them throughout the test period.
3. MVO: This refers to the experimental settings to use the same static expected return estimations as rMVO, but the covariance matrix estimation is dynamic: calculated from the return series in the past 90 days and updated on a daily basis.
4. Dynamic robust MVO (drMVO): This refers to the experimental settings to use the robust covariance matrix estimation as rMVO, but the expected return estimations are updated daily on a sliding window of 30 days.
5. Dynamic MVO (dMVO): This refers to the experimental settings to use the covariance matrix estimation based on a sliding window of 90 days and the expected return estimations on a sliding window of 30 days.

The portfolio performances are examined via trading simulations from 2016-03-09 to 2017-09-30 (579 days in total).[10] Figure 4.6 depicts the growth of capital beginning from 1 dollar. Table 4.2 reports two important metrics for portfolio performance, namely, the compound annual growth rate (CAGR) and Sharpe ratio [78], which is a common risk-adjusted return measure among practitioners. The formulas for calculating the CAGR and the Sharpe ratio are as follows:

$$\text{CAGR} = [(C^{pfl}_{t+\Delta t}/C^{pfl}_{t})^{\frac{365.25}{\Delta t}} - 1] \times 100\% \tag{4.17}$$

$$\text{Sharpe ratio} = \mathbb{E}(\mu^{pfl}_{d}/\mu^{EW}_{d}) \times \sigma(\mu^{EW}_{d})\big/\sigma(\mu^{pfl}_{d}) \tag{4.18}$$

where C^{pfl} denotes the amount of capital for a portfolio; $\mathbb{E}(\cdot)$ denotes the expected value of a random variable; $\sigma(\cdot)$ denotes the standard deviation of a variable.

[10] This time span is roughly chosen because it is reasonable to assume the asset correlations keep the same. If simulation is carried out for a longer period, we have to access historical corpus of the Reuters Company Business Descriptions and Wikipedia pages, which is out of scope for our discussion.

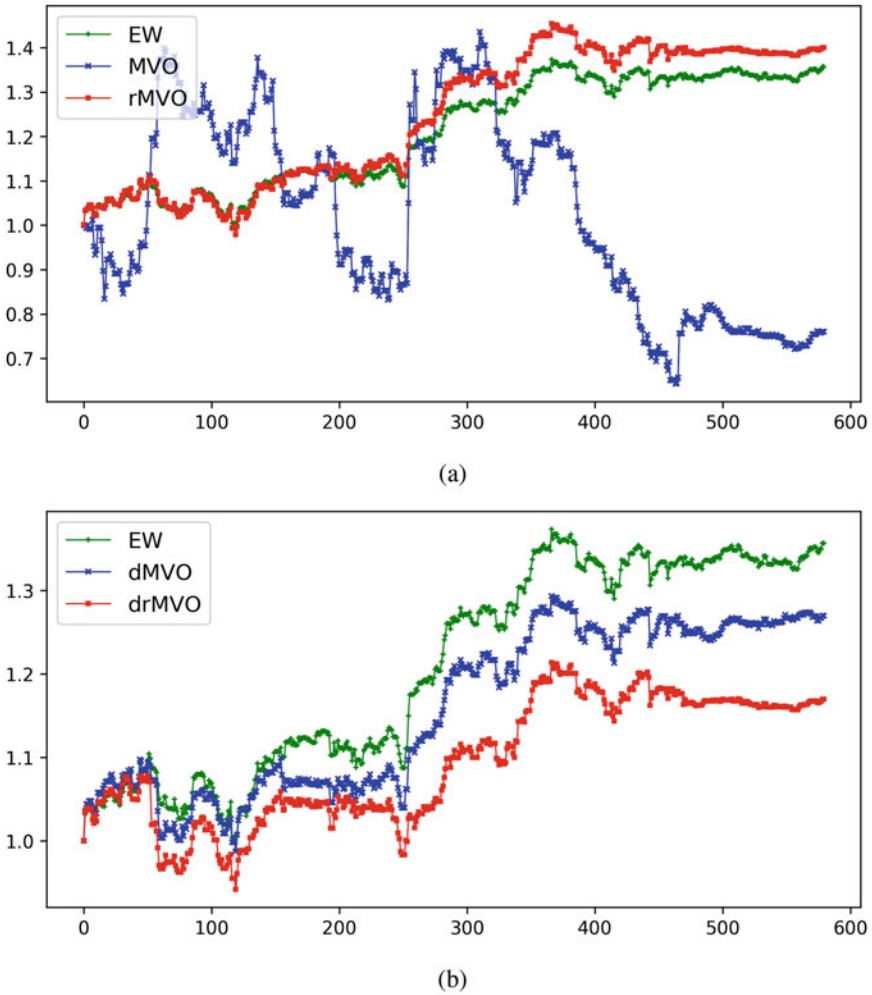

Fig. 4.6 Performance with different experiment settings [191]. (**a**) Single period (static) portfolios. (**b**) Multi-period portfolios, daily rebalancing

Table 4.2 Major statistics of the portfolio performance [191]

Portfolio setting	CAGR(%)	Sharpe ratio
EW	21.21	1.00
MVO	−15.90	−0.20
rMVO	**23.68**	**1.01**
dMVO	16.27	0.78
drMVO	10.39	0.52

In the 579-day simulation, the only experimental setting that consistently outperforms the EW is rMVO. The rest of the portfolios cannot compete with EW even before deducting transaction cost and exhibit significantly higher volatilities. This outcome well addresses the importance of having robust estimators for the mean-variance method, which in our solution is by introducing a semantic vine structure of assets. With only robust estimation of returns, namely, MVO, the portfolio exhibits high volatility and (thus) bad profitability. This could be attributed to two possible reasons: (1) the unstable estimation for the asset covariance matrix and (2) the static return estimations, though robust, may not be accurate.

The fact that both drMVO and dMVO are not performing as well as EW may suggest that dynamic estimation is not very helpful, as long as the process does not involve a robust approach. The capital amount of the two settings move in similar patterns, though dMVO is slightly better than drMVO in terms of expected returns. On the other hand, in terms of volatility, drMVO is slightly more stable than dMVO. However, the differences seem insignificant.

The above observations (not satisfying portfolio performances) may lead to a conclusion that a match of time periods of data used to estimate expected returns and the covariance matrix is important if only numerical data is available.

The experimental results here also resonate with one of our former statement that "the most serious problem of the mean-variance efficient frontier is probably the method's (in-)stability" [183]. The situation of dMVO is that even though it dynamically estimates both expected returns and covariance and rebalances daily with the updated information, the portfolio stays on the "*elusive* efficient frontier" and performs even worse than rMVO [191].

4.4.3 Benchmarking Arbitrary Vines

The experiments from Sect. 4.4.2 have demonstrated the effectiveness of using our invention of semantic vine to robustly estimate the covariance matrix in the mean-variance optimization construction. However, we are not clear if a random vine structure could induce a robust covariance matrix estimation of the same quality. As a further step, we examine the confidence that the semantic vine is superior to many other vine structures. In this experiment, we use the rMVO model setting because it has the best performance among its peers in the discussion of Sect. 4.4.2. We substitute the semantic vine in rMVO with some random vine structures and see how the portfolio performance will be affected. Without knowing the dependence structures, random vines are the state-of-the-art modeling that (still) preserves the robustness of covariance matrix estimation, and there is no evidence that using C-vine or D-vine makes any difference compared to other regular vine structures. Therefore, our benchmarks are valid. To facilitate easy computation of partial correlations, we construct the standard C-vine and D-vine structure and assign random numbers between -1 and 1 to each edge.

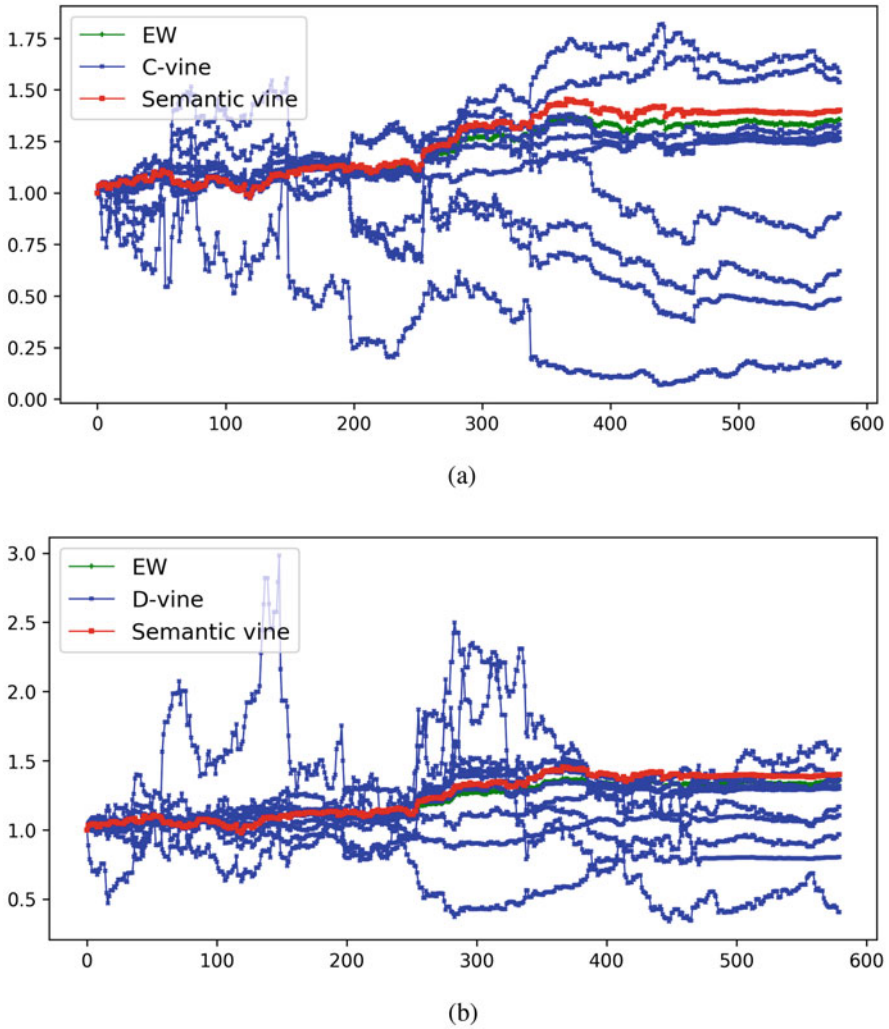

Fig. 4.7 Performance with different vine structures [191]. (**a**) rMVO portfolio with C-vines. (**b**) rMVO portfolio with D-vines

The edge set for standard five-element C-vines is $\mathscr{E}_1$={(1, 2), (1, 3), (1, 4), (1, 5)} $\mathscr{E}_2$ = {(2, 3), (2, 4), (2, 5)} $\mathscr{E}_3$ = {(3, 4), (3, 5)} $\mathscr{E}_4$ = {(4, 5)}. Comparably, the edge set for the D-vines is $\mathscr{E}_1$ = {(1, 2), (2, 3), (3, 4), (4, 5)} $\mathscr{E}_2$ = {(1, 3), (2, 4), (3, 5)} $\mathscr{E}_3$ = {(1, 4), (2, 5)} $\mathscr{E}_4$ = {(1, 5)}. We obtain 20 alternative vines in total. We use the abbreviated notation, for instance, Cv-n for the n-th random C-vine and Dv-n for the n-th random D-vine. Each portfolio performance is illustrated in Fig. 4.7, and statistic measures and significance test results are provided in Tables 4.3 and 4.4, respectively. If we assume the performance measures

Table 4.3 Major statistics of the portfolio performance, those measures better than EW are in bold [191]

Pfl. setting	CAGR(%)	Sharpe ratio	Pfl. setting	CAGR(%)	Sharpe ratio
EW	21.21	1.00	S-vine	**23.68**	**1.01**
Cv-1	−66.31	−0.45	Dv-1	**21.50**	0.58
Cv-2	−36.43	−0.46	Dv-2	**21.45**	0.58
Cv-3	**31.13**	0.90	Dv-3	**33.36**	0.25
Cv-4	−6.38	−0.10	Dv-4	19.05	0.62
Cv-5	15.69	0.60	Dv-5	−12.92	−0.17
Cv-6	17.71	0.79	Dv-6	−1.96	−0.03
Cv-7	15.96	0.51	Dv-7	10.27	0.15
Cv-8	−25.97	−0.42	Dv-8	−43.46	−0.19
Cv-9	**33.81**	0.78	Dv-9	6.50	0.25
Cv-10	19.80	0.88	Dv-10	17.93	0.78

Table 4.4 Significance test of the hypothesis that the semantic vine is superior to an arbitrary C-vine or D-vine [191]

p-value	Chebyshev's bound	Student's t test
Cv-CAGR	0.9405	0.2349
Cv-Sharpe	0.6064	0.0186
Dv-CAGR	0.8278	0.1022
Dv-Sharpe	0.1580	0.0000

are normally distributed, we see a very low likelihood that the better performance of the semantic vine is coincidental, suggested by Student's t. The Sharpe ratio of the semantic vine is significantly higher than either arbitrary C-vines or D-vines. Even in an distribution-agnostic setting, Chebyshev's inequality shows low probability that the performance measures of the semantic vine is drawn from the population of other vine structures.

The trading simulation results presented in Table 4.3 indicate the importance of having a *proper* vine structure. With arbitrarily grown vines, the portfolio does not always outperform the EW, which nullifies the effort to strategically rebalance portfolio weights. Table 4.3 demonstrates that only 2-out-of-10 C-vines and 1-out-of-10 D-vines experimented with provides better portfolio return than the semantic vine in terms of CAGR. However, this is at the expense of significantly higher risk.

None of the experimented arbitrary vine structures outperforms the semantic vine in terms of Sharpe ratio, which means that taking the risk premium is not worthwhile because it actually reduces the expected return for one unit of risk. In fact, none of the experimented arbitrary vine structures outperforms the EW in terms of Sharpe ratio. With a simple calculation based on Table 4.3, the average CAGRs for C-vines and D-vines are −0.1% and 7.2%, respectively, both with a large variance and significantly lower than EW. The average Sharpe ratios are 0.30 and 0.28, which are significantly lower than that of EW (1.00).

Further interpretation of the experimental results reveals that the robust covariance matrix estimation guarantees robust efficient portfolio weights, but does not

necessarily guarantee robust portfolio returns. Because robust portfolio returns, after all, depend on how accurate the rebalancing strategy predicts the return distributions of assets and accordingly mitigates the portfolio risk. We clearly observe that the portfolio returns of arbitrarily grown C-vines and D-vines are not stable. In some of our simulations using arbitrary vines, losing 80% of the principal capital is possible. We also notice that vines of similar CAGR in Table 4.3 may have very different topological structure as well as Sharpe ratios, which suggest that the portfolio return and portfolio risk are not rigidly proportional even when a vine structure is used in robust covariance matrix estimation. Therefore, rebalancing is still beneficial for robust asset allocation.

4.4.4 Model Scalability

Matrix-related computing usually has a high time complexity. For instance, multiplication of two matrices and calculation of the correlation matrix have cubic time complexity. In financial applications, the time complexity of vine-growing and robust correlation matrix estimation algorithms is critical, because as the number of individual assets considered increases, a much more complicated vine structure will be required in limited time.

Our analysis excludes the time for pre-training of semantic space because the update can be done in low frequency and in parallel, while the document vector can be embedded in negligible time after the semantic space is ready. However, the scale of the vine-growing problem also depends on data properties.

For example, the time needed for judging adjacency of edges relates to the average degree of nodes in each layer. A higher average degree indicates more complicated graph structure, and hence more times of judging adjacency will be needed. Consulting some empirical results, we assume that the judging adjacency process can be done in a quasilinear time $\mathcal{O}(n \log n)$. Then, the theoretical complexity for vine-growing would be $\mathcal{O}(n^3 \log n)$, because the robust correlation matrix estimation process has a theoretical complexity of $\mathcal{O}(n^3)$.

Is this time complexity too high for real-world applications? Considering that the naïve calculation of partial correlations already has a complexity of $\mathcal{O}(n^3)$, the semantic vine can be constructed in an acceptable time. The semantic vine-based method would not be significantly slower than computing partial correlations, which is a must for mean-variance optimization.

Table 4.5 reports the CPU times experienced for semantic vine-growing on different numbers of assets and their deviation from the (theoretically) estimated time. The experiments were conducted on a MacBook Pro with 2.6 GHz Intel® Core i5 processors

Another suspicion is, though we have demonstrated the quality of semantic vine constructed on a portfolio of five assets, whether a more complicated semantic vine will retain the same quality and reveal the important correlations between assets as the problem scale increases. As a reply to this concern, we investigate the "Tree 1"

Table 4.5 Comparisons of empirical and theoretical time complexity at different problem scales [191]

Number of assets	5	10	20	50	100	200
CPU time (ms)	1.58	11.8	139	4,320	49,400	758,000
Estimated time (ms)	(1.58)	11.4	182	4,740	49,335	486,000
Error(%)	–	+3.5	−23.6	−8.9	+1.0	+56.0

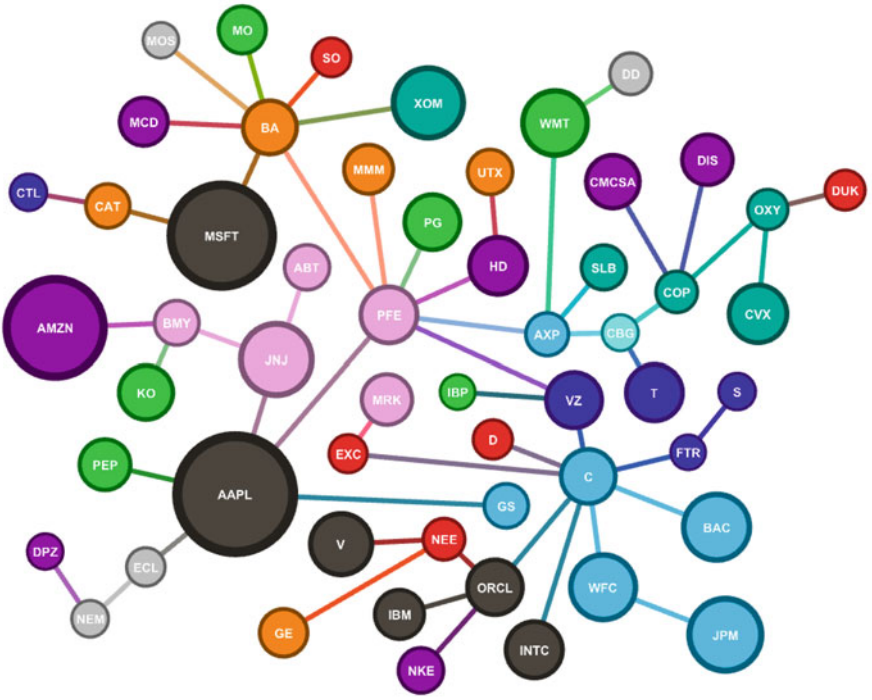

Fig. 4.8 The first layer dependence structure of stocks selected from the US market [191]

on a larger scale of stocks (Fig. 4.8). The sizes of nodes are according to market capitalization; each node is marked by the ticker of that stock. A full list of these stocks and the whole vine structure are provided in Appendix A.

In Fig. 4.8, the stocks are divided into different color groups according to their business sectors suggested by the Global Industry Classification Standard (GICS) and the Thomson Reuters Business Classification (TRBC). The two systems are almost identical in terms of business sectors though differ at more specific levels: the industry and sub-industry codes. At the business sector level, only 2 out of 55 are differently classified. We can observe from Fig. 4.8 that many companies in the same industry are linked, such as Wells Fargo and JPMorgan [31]. Additionally, we observe healthcare clusters, e.g., BMY-JNJ-ABT; banking clusters, e.g., BAC-WFC-JPM; and energy-mining clusters, e.g., COP-OXY-CVX. However, these

companies from the same cluster can have very different surrounding neighbors. For instance, both are classified as consumer discretionary businesses, Comcast is more affiliated with telecom industry, while Amazon is located closer to healthcare and the retailing business [191]. Despite the fact that the semantic vine actually has more layers that are not shown in Fig. 4.8, the first layer dependence structure alone captures more information than the popular industry classification standards since a company is defined by its linked neighbors.

4.5 Summary

Many different types of approaches have recently been developed to incorporate prior knowledge into financial applications. The knowledge types include sentiment knowledge, semantic knowledge, and common sense knowledge. However, not many of these approaches have attached importance to the issue of robustness. One of the important reasons why leveraging such knowledge could bring about improvement is that knowledge eliminates uncertainty and forms a more consistent perspective for us and the way we do things. Some unnecessary cost comes from our fear and we fancy ourselves clever. This chapter excitingly presents a combination of leveraging prior semantic knowledge and theoretically sound robustness, showing potential financial applications of NLP and knowledge representation.

Chapter 5
Sentiment Analysis for View Modeling

But this long run is a misleading guide to current affairs. In the long run we are all dead.

—John M. Keynes

Abstract This chapter investigates a method to incorporate market sentiment to asset allocation models. In the previous chapter, we experimented with robust mean-variance optimization, which is a static process that finds the status quo optimal portfolio weights and surfs market fluctuations. However, an important piece of the jigsaw is missing, i.e., the irrational components in rise and fall of asset prices. In fact, if all the market participants hold the same robust Markowitz portfolio, the market would not clear, nor would transactions happen. The Black-Litterman model provides us an entry to include subjective views to asset allocation models. As an extension to it, concept-level sentiment analysis methods described in this chapter will be used to compute the subjective views, emulating a financial analyst's activities.

Keywords Concept-level sentiment analysis · Subjective view modeling · Market sentiment · The Black-Litterman model · Sentic computing · ECM-LSTM

5.1 Concept-Level Sentiment Analysis

Sentiment analysis is a "suitcase" research problem [25] that requires dealing with many NLP subtasks, such as aspect extraction, concept extraction, named entity recognition, subjectivity detection, and sarcasm detection (see Fig. 5.1). But to integrate and make use of sentiment analysis for broader computational social science, it will be beneficial to include complementary tasks such as personality recognition, user profiling, and multimodal fusion. In financial markets, participants' behavior such as the fear of steep fall, over-confidence on the trend, and risk aversion are all related to sentiment. Consequently, sentiment analysis or affective computing is

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4_5

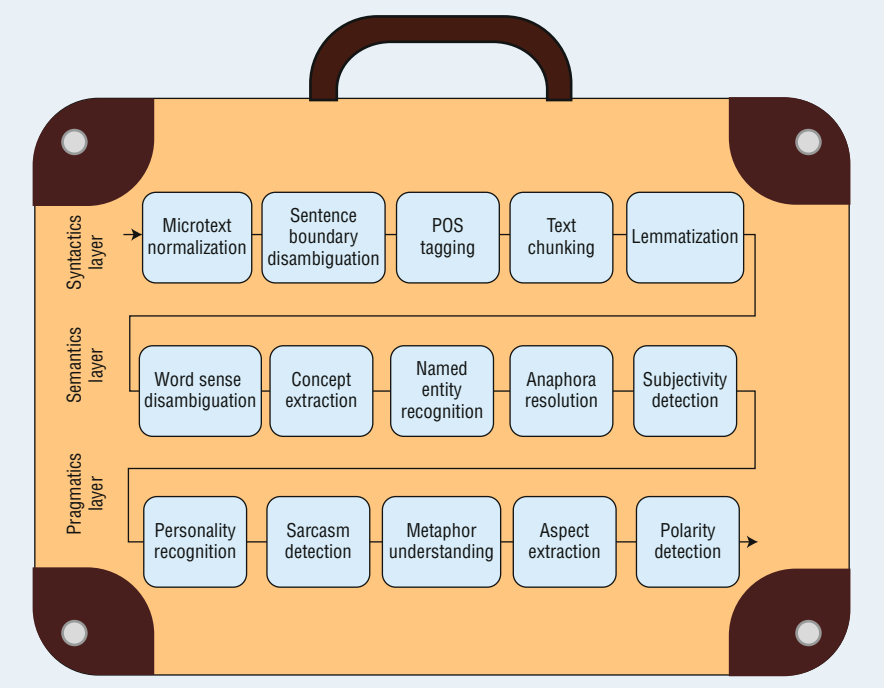

Fig. 5.1 The suitcase metaphor for sentiment analysis [25]

yet another important perspective on financial activities. According to the prophetic five-eras vision of the future web (see Fig. 5.2 of [127]), market sentiment would become a prominent factor that influences trading and information flow as well as shaping products and services. The market behavior has been profoundly changed due to the introduction and evolution of information infrastructure (see discussions by Gagnon and Goyal [80], Jensen [64]).

The sentiment analysis research becomes popular in synergy with the development of Web 2.0. We categorize the research approaches to sentiment analysis and affective computing into three main groups: knowledge-based techniques, statistical methods, and hybrid approaches. Knowledge-based techniques are brainchildren of some ambitious early attempts to build large-scale language resources that can curate the sentiment and relations of every expression. Such projects include the Cyc [69] led by Douglas Lenat, Open Mind Common Sense (OMCS) from which ConceptNet [100] was built, and WordNet [60]. To assign sentiment information for those knowledge bases, there are several competing computational models for sentiment representation based on different psychological theories of emotion [109]. Categorical theories of emotion define a finite set of labels and assign core emotion labels to words, for example, WordNet-Affect [177]. At the top level, sentiment words are either positive or negative according to the primary core emotion.

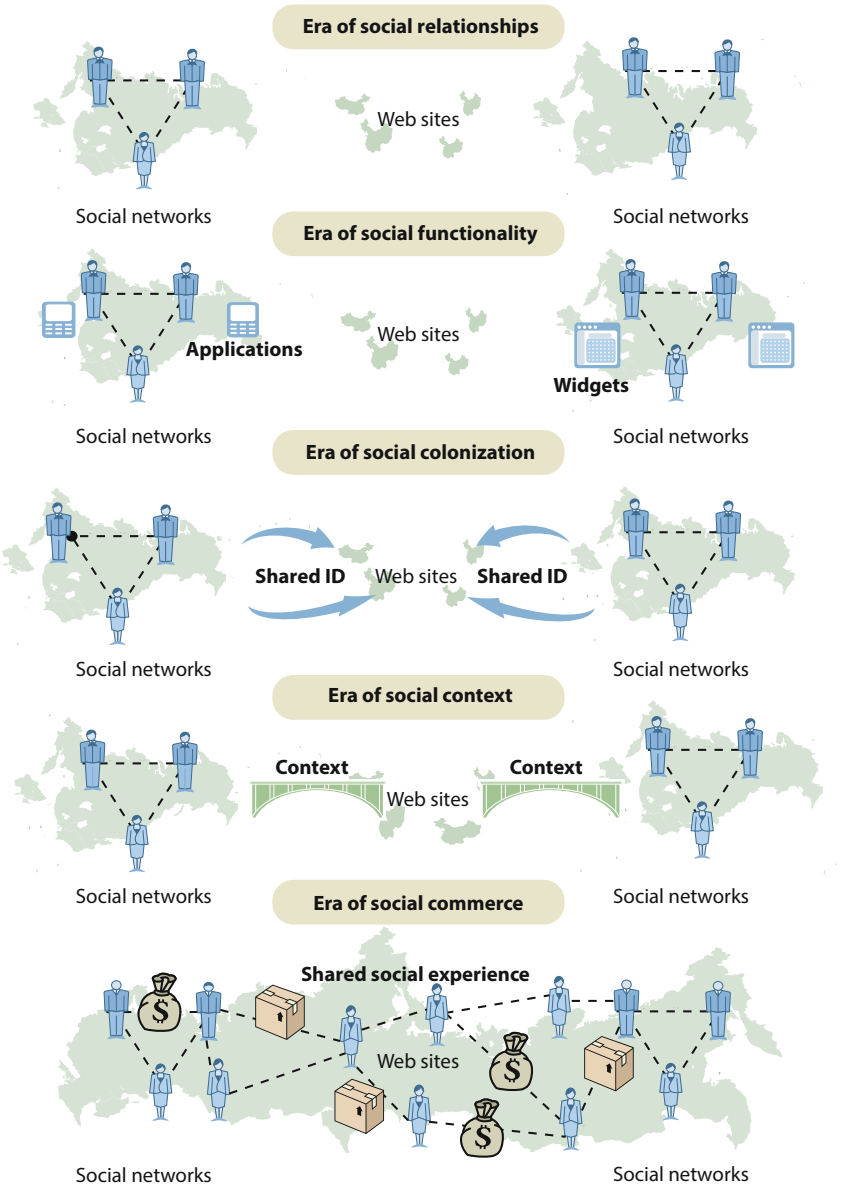

Fig. 5.2 The five-eras vision of the future web [127]

The manual work required here is to categorize a finite set of emotions. Opinion Lexicon [76] is one example of this scheme. At a lower level, emotion labels can also be related to each other. Such models include dimensional activations as basic factors that underlie the labels, e.g., the arousal-valance model, or more information

such as subjectivity and intensity to the knowledge base. SentiWordNet [5] is a good representative of this kind. Another popular open domain resource is SenticNet [28], which contains entries not at the word level, but at the concept level to tackle the problem of phrases and multi-word expressions [24, 144].

Concept-Level Sentiment Analysis (CLSA) make use of a knowledge base where concepts and their sentiment aspects are stored, e.g., in SenticNet. The polarity detection algorithm retrieves concepts as well as the corresponding polarity scores from the knowledge base. The knowledge base can keep expanding and is transparent to users. This is a desirable feature that many financial applications would require. The look-up mechanism of CLSA partially solves the mysterious compositionality of sentiment inside concepts, e.g., "`thirsty`"[1] is negative, but "`thirsty_for_knowledge`" is positive.

5.1.1 Sentiment Analysis in the Financial Domain

Human society has created enormous complex systems, and financial markets are among the most chaotic and dynamic ones. Through bids and offers on the financial assets, many factors could lead to market price fluctuations. The psychology and conduct of market participants have a significant role to play in this price-forming system. Public mood is a highly effective and universal variable that represents market participants' attitudes. Moreover, the growing popularity of social media has made the spread of information faster than ever before. As a result, the subjective views on the market will in some periods dominate the market and bring about irrational market behaviors. A recent study [95] suggests that current stock price movements in the world's major financial markets are essentially influenced by new information and the investors' belief rather than how the business of companies are running.

Several hand-crafted public resources are already widely used to analyze public mood in the financial domain. Some of them are contributed by economics and finance researchers, e.g., the General Inquirer [84], the word list from analyses of tone of earnings press releases by Henry [73], and a more recent Loughran & McDonald word list [103]. Wuthrich and his colleagues have reported in their pioneer work [187] circa four hundreds keyword tuples by consulting financial experts. The tuples contain adjectives such as "high" and "low" and have demonstrated predictive power for market movements. The attempts to build more comprehensive lexicons and word lists in the financial domain automatically have lasted to recent time [70, 165]. These studies have to use label propagation to transfer sentiment information from seed words to more related words. However, many focus overly on the methods, and the final lexicons produced are not made available to the public, such as in [165]. In open domain there are more accessible lexicons with richer

[1] We use typewriter font to denote concepts throughout the book.

sentiment information rather than sentiment polarities. Some of them can be used for financial forecasting as well. For example, SenticNet stores four-dimensional values of the hourglass model [27], which is derived from Plutchik's wheel of emotions. It has been used to analyze the polarity of financial tweets and form sentiment time series [189, 192]. Another quite popular resource among researchers of finance is called Profile of Mood States (POMS). The POMS was developed by psychologists using a rather different sentiment space to scale mood aptitude or subjectivity, compared to Plutchik's wheel of emotions. The original form of POMS [151] proposed in 1983 consists of six factors: tension-anxiety, depression-dejection, anger-hostility, fatigue-inertia, vigor-activity, and confusion-bewilderment. Several revised versions of the POMS, like OpinionFinder [184], adopted the similar factor categories and are used later in some influential works, such as [19, 124]. The six factors are not necessarily independent of each other, and such redundancies are usually useful to accurately model emotion states. Combined with machine learning techniques, application of the lexicons in practice can analyze the sentiment at word level, concept level, sentence level, and paragraph level. For instance, the two versions of Sentic API[2] work at concept and sentence level. The AZFinText system [148] works at a document level. The Stock Sonar [58] is used to conduct sentiment analysis at both the word level and phrase level. In the end, the system would do polarity classification at a document level.

Debates have been there on the effectiveness of using alternative data for financial forecasting. We believe the use of public mood is worthwhile for analyzing financial markets and specific financial assets even for the professionals because it brings in public yet incremental information. On the contrary, technical analysts entirely rely on mining of the historical price patterns, where the information-to-noise ratio is very low. In many trendy AI-based systems, especially those that apply machine learning and deep neural networks to stock market prediction, the community falls into the same situation as technical analysts. The use of large computational power and model complexity cannot exceed the limitation and turn the input data into gold. In fact, financial time series are extremely difficult to forecast. A study by Xing et al. [193] suggests that most complex systems are chaotic, namely, there are no "detectable patterns" even for deterministic systems. As time evolves, small errors in observations quickly lead to different results. This does not necessarily mean that the current prices reflect all the past information as the efficient-market hypothesis (EMH) suggests and there is nothing to learn from lagged values. In one sense, the difficulty lays on accurate observations of the past sequence. In another sense, as the prices are driven by new information, the past patterns fade quickly away. Consequently, the pattern chasers are always one step behind if they simply build the model with past prices and their models mimic the trend with no predictive power.

In addition to the price series, macroeconomic variables are sometimes considered as the driving forces for market fluctuations in the literature as well. The

[2] http://sentic.net/api/

multi-factor models [56] usually take into account a company's book value and investment. Nevertheless, the problem with having macroeconomic variables is that those factors are updated in rather low frequencies. The public mood, in contrast, is easier to sample and access in real-time. Unlike many economic factors, the public mood can be instantaneously monitored and estimated as an aggregation of the market sentiments of individuals. Public moods can be mined from many sources that appeared in previous studies, such as newspapers [187], RSS feeds [201], stock message boards [3], microblogging platforms [157], social media, and more [182]. For example, Zhang and Skiena [201] utilized counts of positive and negative words to derive polarity and subjectivity for certain companies; Antweiler and Frank [3] labeled some messages by hand to train a Naïve Bayes classifier to predict either bullish, bearish, or neutral based on the BOW representation of messages; Smailović [157] used emoticons to classify a big dataset of tweets and trained an SVM based on the collection.

More recently, Weichselbraun and his colleagues [182] proposed to analyze the sentiment of social media streams based on dependency trees that are enriched with sentiment information mined from a knowledge base to include grammatical structures. Although the methods used are diverse from knowledge-based approaches to machine learning, most of them have discovered significant correlations between the public mood and the movement of asset prices. Statistical test and simulation results further assisted to confirm the predictive power of public mood [18, 19].

5.2 Market Views and Market Sentiment

Although the market sentiment is important, it is not adequate to predict the asset prices based solely on a collection of public mood data, let alone to directly make his investment decision for an individual. This is because the underlying mechanism of price formation is complicated: public mood does not directly affect the market; it does indirectly through other market participants' views and their consequent behavior. There may be multiple rounds of interaction, which are called "higher-order beliefs" (the belief that other participants would hold a belief that...) in game theory [8, 119]. Subsequently, a natural question to ask is "how to bridge public mood with market views (of the investor himself)" or, in other words, how to change the perspective from an observer to a decision-making ego. However, we know very little about the mechanism of how market views are formed from public mood especially in the context of asset allocation and invest management. In the rest part of Sect. 5.2, inspired by the Bayes' theorem, we will introduce a method to compute market views from both the historical price series and an investor's prior based on the sentiment time series from the social media. We conduct ablation analyses and find out that using this sentiment prior would on average enhance the annualized portfolio yield by 10% on top of various state-of-the-art asset allocation strategies.

5.2.1 Market Views: Formats and Properties

In Sect. 2.3.2, we introduced the underlying distribution of expected returns by investor's market views as $r_{\text{views}} \sim \mathcal{N}(Q, \Omega)$. This form implies Q as the subjective expected returns and Ω as the variance of r_{views}. Based on the physical meaning of antecedent Q and Ω, the Black-Litterman model defines two types of market views, that is, relative views and absolute views [72]. A relative view takes the form of "I have ω_1 confidence that asset a_x will outperform asset a_y by $b\%$ (in terms of expected return)"; an absolute view takes the form of "I have ω_2 confidence that asset a_z will outperform the market by $c\%$". Therefore, we arrive at the mathematical definition of market views (matrices) as follows.

Definition 5.1 For a portfolio consisting of n assets, a set of k views can be represented by three matrices $P_{k,n}$, $Q_{k,1}$, and $\Omega_{k,k}$. P indicates the assets mentioned in views. The sum of each row of P should either be 0 (for relative views) or 1 (for absolute views); Q is the expected return for each view; and the confidence matrix Ω is a measure of covariance between the views.

We write the confidence matrix as $\Omega = diag(\omega_1, \omega_2, \ldots, \omega_n)$. This is possible because the market views are assumed to be independent of each other by the Black-Litterman model. Consequently, the confidence matrix is full rank. In fact, as long as the k views are compatible (not self-contradictory), which is a prerequisite for further computation, the diagonal assumption of the confidence matrix will not harm the expressiveness of the set of market views. Suppose we have a counterexample when the confidence matrix $\Omega_{k,k}$ is not diagonal, then we do spectral decomposition to Ω: $\Omega = V\Omega^{\Lambda}V^{-1}$, where Ω^{Λ} is by definition diagonal. In such case, we can appoint Ω^{Λ} to be the new confidence matrix and let the new mentioning matrix and the new expected return matrix be $P^{\Lambda} = V^{-1}P$, $Q^{\Lambda} = V^{-1}Q$. Under Definition 5.1, we describe two crucial properties (Theorem 5.1 and 5.2) of the view matrices P, Q, and Ω with proofs [188].

Theorem 5.1 (Compatibility of Independent Views) *Any set of independent views are compatible.*

Proof Assume there is one pair of incompatible views formalized as $\{p, q\}$ and $\{p, q'\}$, where $q \neq q'$. Both views are either explicitly stated or can be derived from a set of k views. Hence, there exist two different linear combinations, such that:

$$\sum_{i=1}^{k} a_i p_i = p \qquad \sum_{i=1}^{k} a_i q_i = q \tag{5.1a}$$

$$\sum_{i=1}^{k} b_i p_i = p \qquad \sum_{i=1}^{k} b_i q_i = q' \tag{5.1b}$$

where $(a_i - b_i)$ are not all zeros.

Thus, we have $\sum_{i=1}^{k}(a_i - b_i)p_i = \mathbf{0}$, which means that the k views are not independent. According to the law of contrapositive, the statement "all independent view sets are compatible" is true. □

Theorem 5.2 (Universality of Absolute View Matrix) *Any set of independent relative and absolute views can be expressed with a non-singular absolute view matrix.*

Proof Assume a matrix P mentioning k views in total: r relative views and $(k - r)$ absolute views, that is, $P_{k,n} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{r,1} & p_{r,2} & \cdots & p_{r,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,n} \end{bmatrix}$.

Correspondingly, the return vector is $Q_{k,1} = (q_1, q_2, \ldots, q_k)$, the capital weight vector for assets is $\mathbf{w} = (w_1, w_2, \ldots, w_k)$.

An equivalent expression of the same views should have the same expected returns for each asset regardless how many times and where they are mentioned. Hence, we can write $(r + 1)$ equations with regard to r new variables $\{q'_1, q'_2, \ldots, q'_r\}$, where $j = 1, 2, \ldots, r$:

$$1 + q'_j = \sum_{i \neq j}^{r}(1 + q'_i)\frac{w_i}{\sum_{s \neq j} w_s}(1 + q_j) \tag{5.2a}$$

$$\sum_{i=1}^{r} q'_i w_i + \sum_{i=r+1}^{k} q_i w_i = Q\mathbf{w}^{\mathsf{T}} \tag{5.2b}$$

We consider asset $\{a_{r+1}, a_{r+2}, \ldots, a_k\}$ to be *one* compound asset, then, the return of this compound asset is decided by $P_{r,n}$. Hence, r out of the $(r + 1)$ equations above are independent.

There must exist a unique solution with the form of $Q' = (q'_1, q'_2, \ldots, q'_r, q_{r+1}, \ldots, q_k)$ to the aforementioned $(r + 1)$ equations, according to Cramer's rule, such that view matrices $\{P', Q'\}$ are equivalent to view matrices $\{P, Q\}$ for all the assets considered, where

$$P'_{k,n} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & p_{r,r} = 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,n} \end{bmatrix}.$$

Obviously, $P'_{k,n}$ only consists of absolute views. By deleting dependent views from $P'_{k,n}$, we can have a non-singular matrix that only consists of absolute views and is compatible. □

Providing Theorem 5.1 and 5.2, without loss of generality, we are able to impose one equivalent yet more strict definition of market views for the purpose of reducing the computational complexity. In practice, Definition 5.2 is more frequently used, because the elimination of matrix P makes the definition concise and easy to understand. In fact, the format of asset mentioning matrix in Definition 5.2 not only facilitates the restriction of absolute market views but also provides independency. This guarantees that the Black-Litterman assumption, which says the market views can be represented with a multivariate normal distribution, is mathematically sound.

Definition 5.2 Market views on n assets can be represented by three matrices $P_{n,n}$, $Q_{n,1}$, and $\Omega_{n,n}$, where $P_{n,n}$ is an identity matrix ($\mathbb{I}$); $Q_{n,1} \in \mathbb{R}^n$; $\Omega_{n,n}$ is a nonnegative diagonal matrix.

Furthermore, following the steps described in [146], a specification of equation 2.12 can be derived from equation 2.13 and Definition 5.1 that:

$$\mu_{BL} = [(\tau\Sigma)^{-1} + P'\hat{\Omega}^{-1}P]^{-1}[(\tau\Sigma)^{-1}\Pi + P'\hat{\Omega}^{-1}Q] \tag{5.3}$$

$$\Sigma_{BL} = \Sigma + [(\tau\Sigma)^{-1} + P'\hat{\Omega}^{-1}P]^{-1} \tag{5.4}$$

Therefore, the task of computing market views can be redescribed as to estimate the variables (including P, Q, and Ω, while others are considered given by the CAPM) in Equations 5.3 and 5.4 with the assistance of a sentiment prior.

5.2.2 *Estimating Volatility, Confidence, and Return*

For the equilibrium risk premiums Π, we use the calculation suggested by CAPM (equation 2.11). That is, a premium proportional to the realized volatility calculated from historical price series. Then to estimate the parameters of the posterior (Gaussian) distribution of the expected portfolio returns, three variables are to be determined as in the Black-Litterman model: the equilibrium volatility as a covariance matrix (Σ), the investor's confidence of his own views (Ω), and the investor's expected returns as in his views (Q).

We first use the traditional truncated method to calculate the covariance matrix instead of the state-of-the-art method introduced in Chap. 4 for the fairness of comparison. According to the recommendation of the Black-Litterman model, the past k-days observed returns are used for asset pairs (a_i, a_j). The element σ_{ij} as in covariance matrix Σ is estimated as below:

$$\hat{\sigma_{ij}} = k^{-1} \sum_{n=1}^{k} (R_{i,-n} \cdot R_{j,-n}) - k^{-2} \sum_{n=1}^{k} R_{i,-n} \sum_{n=1}^{k} R_{j,-n} \tag{5.5}$$

where $R_{i,-n}$ is the return of asset a_i on the n-th past day.

The classical form of the Black-Litterman model [14] relies on investing experts to manually set the confidence matrix Ω based on their own experience. At the worst cases, where the investor has no idea how to derive the confidence matrix, a numerical example provided by [72] pointed out a primary estimation:

$$\hat{\Omega} = diag(P(\tau\Sigma)P') \tag{5.6}$$

We give the explanation for this estimation as follows. Because Σ is by definition a covariance matrix, $P(\tau\Sigma)P'$ can also be understood as $cov(\tau P\Sigma, \tau P\Sigma)$, which is a covariance matrix of the expected returns in the views. Note that the mentioning matrix P "filters out" the covariances not relevant to the views. With Definition 5.2, where P is an identity matrix, this estimation is more understandable. Because $P(\tau\Sigma)P'$ is already diagonal, the latent hypothesis here is that the variance of an absolute view on asset a_i is proportional to the volatility of asset a_i. This hypothesis shares the same idea as the CAPM: not only the risk premium comes from volatility, but also the confidence of any judgment would decrease the same amount if the return is more volatile. In the example by [72], the estimation of Ω utilizes only the past information of asset price volatilities.

Compared to volatility, the expected return has a more directly perceivable relation to the market sentiment. In contrast to the naive assumption that positive market sentiment leads to positive returns and vice versa, our assumption here is more developed. We believe there exists a strategy that "responds to the market sentiment" and can surf the market and statistically makes profits (generates alpha). However, such a strategy can be complicated. Therefore, we employ machine learning techniques to "learn" this strategy under the framework of the Black-Litterman model. That is, imagine an agent who empirically forms and updates their views using information like the past price series ($\boldsymbol{\pi_{t,k}}$) and trading volumes ($\mathbf{v_{t,k}}$). In our extension, these activities further involve a new prior: sentiment time series derived from the alternative data stream obtained from the social media. We denote this new prior by $\mathbb{S}_\mathbf{t}$. Now the problem (formally) becomes learning a proper function $\mathbb{F}$ that maps the expected return estimation to each time period t:

$$\hat{Q}_t = \mathbb{F}[\hat{Q}_{BL}(\boldsymbol{\pi_{t,k}}, \mathbf{v_{t,k}}, \mathbb{S}_\mathbf{t});\ Q_t^*] \tag{5.7}$$

where Q_t^* is the supervision.

We compare three implementations of the function $\mathbb{F}$, which are neural-fuzzy approach (DENFIS), deep learning approach (LSTM), and an innovative deep recurrent neural network design that is based on evolving clustering method (ECM) and LSTM architecture, hence termed ECM-LSTM. The three implementations are chosen for the motivation that we want to have a good coverage of various types

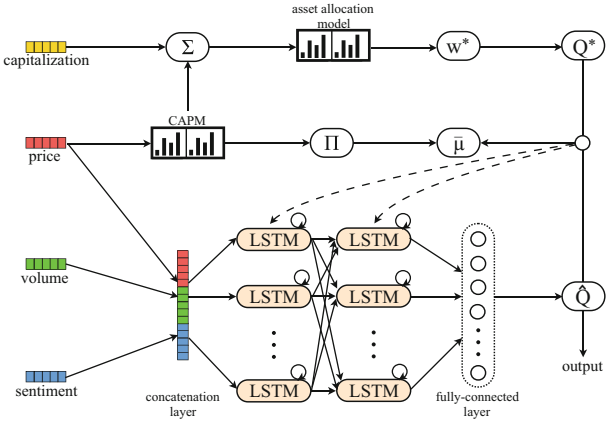

Fig. 5.3 Model training process for generating market views

of neutral network-based approaches. Figure 5.3 illustrates the dataflow when $\mathbb{F}$ is implemented by LSTM [188].

5.2.3 *DENFIS, LSTM, and ECM-LSTM*

In this section, we introduce the details of ECM-LSTM and show the combination of architecture for both DENFIS and LSTM.

DENFIS is a type of fuzzy inference system (FIS) with neuro-fuzzy rule nodes proposed by Kasabov and colleagues in [83]. The model has a fast adaptive ability because it actively monitors the ever-changing distribution of incoming data and partitions the rule nodes dynamically, so that every-time the activated neurons are different. This feature is especially useful for financial time series. Empirical results show that DENFIS, compared to several other types of fuzzy neural networks, such as Artificial Neuro-Fuzzy Inference System (ANFIS) and Neural Fuzzy Controller (NEFCON), has a better performance in nonlinear chaotic system modeling [193]. We consider the financial market as a complex system in real-world; thus, DENFIS is the most promising model in modeling its dynamics. Each DENFIS neuron has an "IF-THEN" format rule:

$$\begin{aligned} IF \quad & L^{0\sim k}[\pi, v, \mathbb{S}]_{t,i} = pattern_i,\ i = 1, 2, \ldots, N \\ THEN\ & \hat{Q}_t = f_{1,2,\ldots,N}([\pi, v, \mathbb{S}]_t), \end{aligned}$$

where π, v, $\mathbb{S}$ are three attributes and some of the $(2^N - 1)$ functions are activated. L is the lag operator and the pattern parameters are learned online. The rule format is called Takagi-Sugeno-Kang type (TSK) rule for that the output is a linear function

of degrees of applicability for each pattern. We implement the DENFIS model with all the membership functions set up to symmetrical and triangular, that is, two parameters define the activation range of d: at $x = z$ the membership degree equals to 1 and $x = z \pm d/2$ the membership degree equals to 0. Note that for each time period t, z is updated with a linear least-square estimator of existing consequent function coefficients, so as to align the activation of each fuzzy rule.

LSTM is yet another popular model for sequential modeling in recent time. The model equips the simple recurrent neural network with three types of gated units. The gates control information from previous time steps, so LSTM architecture is claimed to perform well in predicting time series with an unknown size of lags and long-term dependencies [75]. Previous studies like [65] also applied LSTM to time series prediction. Other attempts to develop more powerful recurrent neural networks include GRU [48] and some automatically designed variants, though their performance across different tasks are similar [67]. For this reason, we employ the vanilla LSTM unit structure and implement the "$\pi v\mathbb{S}$-LSTM", where inside the neuron cells the update rules are as below for the input gate, forget gate, and output gate, respectively.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [\, h_{t-1}, [\pi, v, \mathbb{S}]_t \,] + b_i) \\ f_t &= \sigma(W_f \cdot [\, h_{t-1}, [\pi, v, \mathbb{S}]_t \,] + b_f) \\ o_t &= \sigma(W_o \cdot [\, h_{t-1}, [\pi, v, \mathbb{S}]_t \,] + b_o) \end{aligned} \tag{5.8}$$

where the sigmoid function is denoted by σ, h_{t-1} is the hidden state of the previous time step, $W_{()}$ denotes the state transfer matrices, and $b_{()}$ is the bias. The rules that update the cell state of each LSTM unit are therefore:

$$\begin{aligned} c_t &= f_t \odot c_{t-1} + i_t \odot (W_c \cdot [\, h_{t-1}, [\pi, v, \mathbb{S}]_t \,] + b_c) \\ h_{t-1} &= o_t \odot \tanh(c_{t-1}) \end{aligned} \tag{5.9}$$

Figure 5.4 shows the mathematical operations inside one LSTM cell as in Fig. 5.3. The model is not trained on a batch of data points and locked static; instead, it is trained in an "online fashion." This means that whenever a new input is received, it goes to the training set. The previous states and parameters of LSTM are used as a pre-trained model, and the LSTM model online has one training data for each period t.

ECM-LSTM is a combination of the ECM mechanism and the LSTM neural architecture proposed in [189]. Inspired by the predictive behavior of a simple LSTM, we find that incoming data often "drives" the forecasts to the opposite directions even if in a longer span the ground truth level remains stable. We interpret this type of movement as noise, that is, noise in the inputs is amplified. It is already confirmed that noise is ubiquitous in real-world financial time series. Considering this fact, over-fitting to meaningless signals is inevitable and will increase errors. To this end, we must find out an approach to learn "not to learn" from the noise.

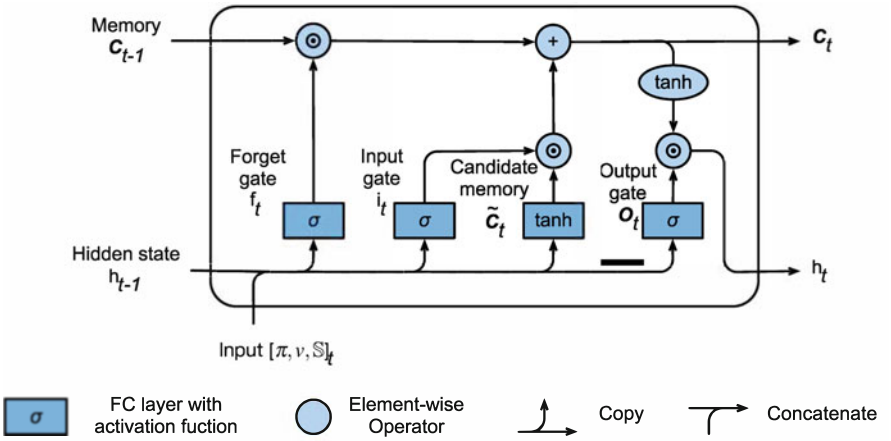

Fig. 5.4 Operations inside a LSTM cell

Our solution is to use the ECM mechanism to filter out less important data. The method was used originally for input space partitioning for rule induction in fuzzy inference systems, by means of dynamically recording and updating the centroids and clustering radii. The method has several good properties, e.g., it is fast because only one round of maximum distance-based clustering without any optimization is required. To take advantage of it, we stabilize the LSTM behavior by learning only from the critical new incoming data and omitting the rest. The criterion is the new incoming data is less important when the old clustering pattern keeps unchanged.

The training and predicting processes of ECM-LSTM are detailed in Algorithm 5.1, where $\hat{Q}_{t-1}$ is the last forecast made by the model, while Q^*_{t-1} is the last observable ground truth. Q^*_{t-1} is computable by inverse engineering the optimal values to maximize the portfolio's return in the last period. We use i, f, and o to represent the activation functions of input gate, forget gate, and output gate, respectively. As in common recurrent neural networks, for time period t, the state for LSTM cells, namely, c_t, is updated by two resources: the current information $[\pi, v, \mathbb{S}]_t$ and the previous state c_{t-1}. At the same time, the ECM mechanism also keeps record of clustering centroids and corresponding radii pairs $(\mathscr{C}_i, \mathscr{R}_i)$ for the input data.

5.2.4 The Optimal Market Sentiment Views

Our task according to Definition 5.2, is to compute the optimal market views $[P, Q^*, \hat{\Omega}]$ or mainly Q^* based on the inverse optimization problem of the Black-Litterman model and sentiment-conditioned return distributions as inputs. In a multi-period model of an asset portfolio where the amount of capital has no memory,

Algorithm 5.1: ECM-LSTM training and forecasting procedure

Data: Incoming data stream π, v, $\mathbb{S}$
Result: Expected return estimation $\hat{Q}_t$
Initialize LSTM parameters W, b;
if $\mathscr{C} = \emptyset$ **then**
  $\mathscr{C}_0 = (\pi_{t,k}, v_{t,k}, \mathbb{S}_t)$;
  $\mathscr{R}_0 = 0$;
  Go to line 15;
else
  $D_{\min} = \min(||(\pi_{t,k}, v_{t,k}, \mathbb{S}_t) - \mathscr{C}_i||)$;
  if $\nexists\ (\mathscr{R}_i \geq D_{\min})$ **then**
    Add $(\pi_{t,k}, v_{t,k}, \mathbb{S}_t)$ to $\mathscr{C}_i$ where $D_{\min}$ holds;
    Go to line 24;
  else
    $(S_{\min}, i) = \min(||(\pi_{t,k}, v_{t,k}, \mathbb{S}_t) - \mathscr{C}_i|| + \mathscr{R}_i)$;
    if $S_{\min} > 2\mathscr{R}_i$ **then**
      Add $(\pi_{t,k}, v_{t,k}, \mathbb{S}_t)$ to $\mathscr{C}$;
      $i_t = \sigma(W_i \cdot [\ \hat{Q}_{t-1}, \pi_{t,k}, v_{t,k}, \mathbb{S}_t\ ] + b_i)$;
      $f_t = \sigma(W_f \cdot [\ \hat{Q}_{t-1}, \pi_{t,k}, v_{t,k}, \mathbb{S}_t\ ] + b_f)$;
      $o_t = \sigma(W_o \cdot [\ \hat{Q}_{t-1}, \pi_{t,k}, v_{t,k}, \mathbb{S}_t\ ] + b_o)$;
      $c_t = f_t \odot c_{t-1} + i_t \odot (W_c \cdot [\ \hat{Q}_{t-1}, \pi_{t,k}, v_{t,k}, \mathbb{S}_t\ ] + b_c)$;
      $\hat{Q}_t = o_t \odot \tanh(c_t)$;
      Update W, b with $\frac{\partial(Q^*_{t-1} - \hat{Q}_{t-1})}{\partial\{i,f,o\}_{t-1}}$;
    else
      Add $(\pi_{t,k}, v_{t,k}, \mathbb{S}_t)$ to $\mathscr{C}_i$ where $S_{\min}$ holds;
      Update $(\mathscr{C}_i, \mathscr{R}_i)$;
      $\hat{Q}_t = o_t \odot \tanh(c_{t-1})$;
    end
  end
end
return $\hat{Q}_t$;

the aim is to maximize the amount of capital in the near future, at period $(t+1)$. We write this optimization problem as:

$$\max_{\mathbf{w_t}} \ C_{t+1} = C_t \times \mathbf{w}_t \odot \frac{\boldsymbol{\pi}_{t+1}}{\boldsymbol{\pi}_t}. \tag{5.10}$$

Since C_t is actually pre-given, the value of variable $\mathbf{w}_t$ is independent from C_t. Therefore, the optimal portfolio weights for each period t are:

$$\mathbf{w}_t^* = \text{argmax} \ \ \mathbf{w}_t \oslash \boldsymbol{\pi}_t \odot \boldsymbol{\pi}_{t+1} \tag{5.11}$$

where $\oslash$ and $\odot$ are element-wise division and element-wise multiplication operators.

Apparently, with the constraint that $\mathbf{w}_t \in [0,1]^n$, the solution of portfolio weights (equation 5.11) would be a one-hot vector representation. In this vector the forecasted one-period returns determine the position index of value 1: the weight of the asset with the maximum price leap ($\frac{\pi_{i,t+1}}{\pi_{i,t}}$) equals to 1.

The real-world interpretation is that one should reinvest his/her whole capital daily to the fastest-growing asset in the next period, providing that there are no short selling and transaction fees. If we set the optimized weight $\mathbf{w}_t^*$ to be $\mathbf{w}_{BL}^*$ in equation 2.14, we derive a multi-period equation 5.12:

$$\mathbf{w}_t^* = (\delta \Sigma_{BL,t})^{-1} \mu_{BL,t} \tag{5.12}$$

Substituting $\Sigma_{BL,t}$ and $\mu_{BL,t}$ with equations 5.3 and 5.4 for each period t, we can have:

$$\mathbf{w}_t^* = [\delta(\Sigma_t + \mathbf{H})]^{-1}\mathbf{H}[(\tau\Sigma_t)^{-1}\Pi_t + P'\hat{\Omega}_t^{-1}Q_t^*] \tag{5.13}$$

where $\mathbf{H} = [(\tau\Sigma_t)^{-1} + P'\hat{\Omega}_t^{-1}P]^{-1}$.

In equation 5.13, $\mathbf{w}_t^*$ is known from the daily price movement of assets. Therefore, our goal is to inversely solve the optimal expected returns in market views Q_t^* for each period t. This result (equation 5.14) has been reported in [189].

$$\begin{aligned} Q_t^* &= \hat{\Omega}_t[\delta[(\tau\Sigma_t)^{-1}+P'\hat{\Omega}_t^{-1}P][\Sigma_t+[(\tau\Sigma_t)^{-1}+P'\hat{\Omega}_t^{-1}P]^{-1}]\mathbf{w}_t^* - (\tau\Sigma_t)^{-1}\Pi_t] \\ &= \delta[\hat{\Omega}_t(\tau\Sigma_t)^{-1} + \mathbb{I}][\Sigma_t + [(\tau\Sigma_t)^{-1} + \hat{\Omega}_t^{-1}]^{-1}]\mathbf{w}_t^* - \hat{\Omega}_t(\tau\Sigma_t)^{-1}\Pi_t \end{aligned} \tag{5.14}$$

5.3 Market Sentiment Computing

The market views rely on computing sentiment time series $\mathbb{S}_t$ in equation 5.7, which itself is an abstract variable that requires NLP and sentiment analysis on a great amount of textual data. The quality of $\mathbb{S}_t$ is no doubt critical, because the data is later employed as Π_t in equation 5.14, where $\hat{Q}_t$ is estimated and trained to minimize the difference. To ensure the quality of $\mathbb{S}_t$ and classify sentiment expressions accurately, we employ the interesting idea of sentic computing [26]. It enables sentiment analysis of text not only at a document or paragraph level but also at the sentence, clause, and concept level. Compared to other machine learning-based sentiment analysis methods, sentic computing combines syntactic feature and knowledge base to do polarity inference and back it up with a learned classifier. A naive statistical sentiment analysis method counts the positive and negative words in a sentence and calculates the net polarity; however, in such a situation, the grammar is not taken into account. By averaging the word polarities, positive and negative words will nullify each other, which brings about difficulties for analyzing sentiment

in complicated contexts [189]. In the remainder of Sect. 5.3, we will describe the Hourglass of Emotions, SenticNet, and how they are used for sentic computing.

5.3.1 The Hourglass of Emotions and SenticNet

Human emotions are more complicated than a sentiment orientation. The nuances between happy, joy, trust, ecstasy, etc. comprise a continuous space of emotions. The Hourglass of Emotions is such a model for affective states classification derived from Plutchik's studies of emotions [133] that depicts this emotion space (see Fig. 5.5). The model assumes that the full spectrum of human emotions can be organized according to four independent but concomitant dimensions: Pleasantness, Attention, Sensitivity, and Aptitude. For example, *Joy* is a mid-level activation of Pleasantness, while *Sadness* is a mid-level inhibition of Pleasantness. More advanced emotions can mingle different activation levels on these four dimensions. For instance, *Joy* + *Trust* + *Anger* = *Jealousy*, which means *Jealousy* can be represented by a quadruple of activation levels $[G(0.5), G(0), G(0.5), G(0.5)]$ termed "sentic vector". Bell curve function $G(x) = -\frac{1}{\sigma\sqrt{2\pi}}e^{-x^2/2\sigma^2}$ maps components of sentic vectors closer to 1.

One advantage of the Hourglass model is that it provides richer information on emotion states associated with words or concepts of natural language, thus enabling computation on multiple dimensions. Meanwhile, a sentiment polarity score between -1 and 1 can be calculated from aggregation of these four factors:

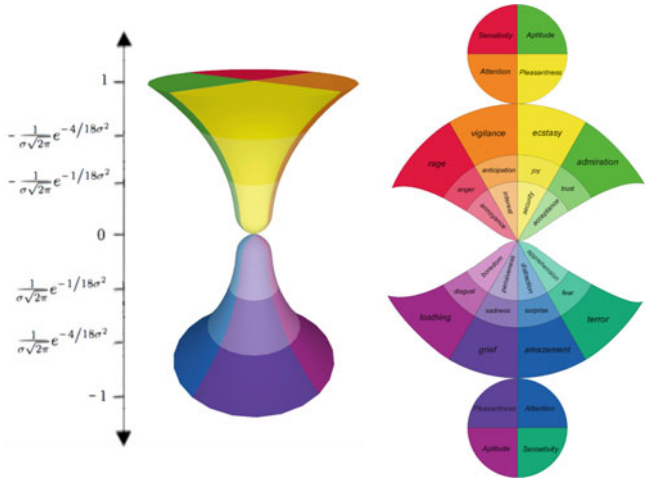

Fig. 5.5 The 3D model of the Hourglass of Emotions [27]

$$\gamma(x) = \frac{Pleasantness(x) + |Attention(x)| - |Sensitivity(x)| + Aptitude(x)}{3} \tag{5.15}$$

SenticNet is a public available knowledge base that stores both semantics and sentics of natural language concepts. The semantics are represented by associated concepts, such as hypernym, hyponym, and related concepts; the sentics, on the other hand, are stored as sentic vectors of the hourglass model. Early versions of SenticNet leverage multiple sources of commonsense knowledge such as OMCS [100, 160] (where ConceptNet is built from), WNA [162], and GECKA [29]. In particular, the concepts are embedded into a space where similar concepts have closer distance. After that, the 24 seed affective concepts in the hourglass model are selected as "centroid concepts." Depending on the relative distances, the activation levels of the seed concepts are decayed and propagated to the concepts belonging to the same cluster. Finally, based on the four activation levels of each concept, equation 5.15 is used to derive the polarity score.

The latest release SenticNet 5 [28] contains over 100,000 concepts. Unlike the previous versions, the concepts are not identically treated, but organized into a multi-layered semantic network that links name entities as well. Based on the contextual features of concepts extracted by a bi-directional LSTM, similar concepts are clustered. Later, sub-component words with the highest occurrence frequencies are selected to form conceptual primitives. For example, the primitive "`eat_food`" leads many more specific concepts such as "`munch_toast`", "`slurp_noodles`," etc. This finite number of primitives solves the thorny coverage issue of commonsense knowledge base. Therefore, we no longer need polarity scores for every concept, and sentic computing can be conducted at the primitive level.

5.3.2 Augmented Sentic Computing

Sentic computing mainly leverages SenticNet [28] as the commonsense knowledge base and sentic patterns [136] as a method to do polarity inference. The patterns are a set of linguistic rules that enables long-term dependency discovery, for example, two words are distant in the sentence but linked by grammatical relations. To infer the overall sentiment polarity, sentic computing first extracts multiple relation tuples from the sentence with the Stanford-typed dependency parser [46]. Meanwhile, a semantic parser traverses each unigram and bigram[3] and attempts to look up the polarity score from a concept-level sentiment knowledge base. Finally, these concepts trigger sentic patterns to process the relations and associated intrinsic polarities. If the concepts are not in the knowledge base, e.g., SenticNet, the method resorts to a classifier built by machine learning. Extending the basic version [134], augmented sentic computing implements modificatory functions instead of polarity

[3] Only meaningful parts of speech tag pairs, such as `ADJ+NOUN`, `VERB+NOUN`, and `VERB+ADV`, are considered and lemmatized.

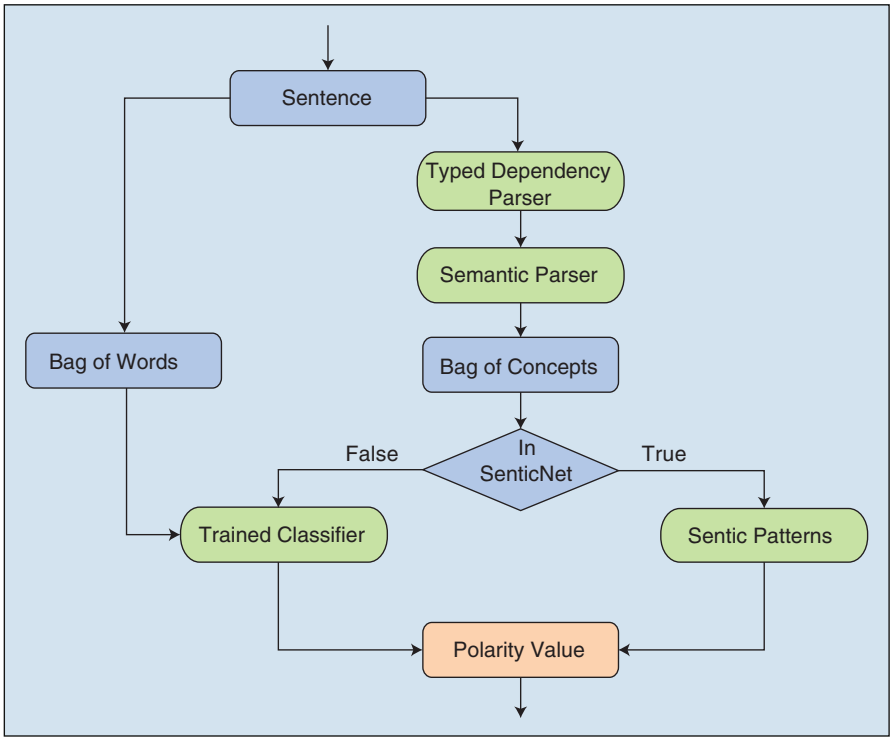

Fig. 5.6 The sentic computing algorithm working at sentence level [189]

algebra for different pivot types. For example, parabolic and power functions are used for decaying and amplifying sentiment intensity.

Figure 5.6 as from [189] depicts this sentence-level polarity detection process.

Augmented sentic computing is powerful in many tricky cases. In the remainder of this chapter, we use augmented sentic computing. Next, we will provide examples of real-world texts from social media where augmented sentic computing outperforms sentic computing and other machine learning-based techniques, for instance, the Google Cloud API for natural language processing.

5.3.3 Examples of Applying Augmented Sentic Patterns

Example 1 I had a feeling $AAPL would go down, but this is stupid

The first phase of analyzing Example 1 is preprocessing: link the cashtag "$AAPL" to "Apple company," and add a period at the end of the complete sentence. The interesting point of Example 1 is that it falls into the category of complicated sentence structure. Not all of the adjectives are describing the target in discussion. Although "stupid" is a negative word, it describes one part of the sentence instead

of "$AAPL." By denying his own previous opinion, the speaker actually means that "$AAPL" would go up and thus expresses a positive mood for the Apple company. For this reason, the user labeled this example as bullish. Imagine a bag-of-words model, which finds two negative words "down" and "stupid": the whole sentence will thus be considered as strong negative. Machine learning-based sentiment analysis models, for instance, Google Cloud Natural Language API (Google SA),[4] may also fail for this example.[5] This result suggests that the syntactic feature of the sentence is not effectively captured by such models.

Augmented sentic computing, in contrast, will first look up the concept "`go_down`" and "`stupid`" from SenticNet. By extracting "`go_down`" as a whole, the model better understands the sentiment of the entire sentence. Although both the two concepts have negative sentiment scores of -0.07 and -0.93, their roles in the sentence is different. Multi-word expression "`this_is_stupid`" contains concept "`stupid`" while the remaining part "`this_is`" carries no sentiment. Consequently, the negative score of -0.93 is inherited through a nominal subject (nsubj) relation. In parallel, another relative clause `feeling_(that)_$AAPL_would_go_down` not only inherits the polarity from concept "`go_down`" but also amplifies the sentiment intensity because of the "acl:relcl" relation. Therefore, the sentiment score would be $-\sqrt{|-0.07|} = -0.27$. At a high level, the two structures are associated by an adversative but-conjunction (but-conj). Thus the sentic patterns are triggered and final polarity score is $\sqrt{|[(-0.27)+(-0.93)]/2|} = +0.78$. This score is passed to the root of the sentence since other components are neutral. The described process is backed up with an alternative multi-layered perceptron classifier to mitigate the coverage problem. When the whole sentence does not contain any concept from the knowledge base, we would not conclude the sentiment to be neutral. Instead, the sentiment score comes from supervised learning. Finally, if the message contains more than one sentence, we will conduct augmented sentic computing for each sentence and average the sentiment scores as an overall result. Figure 5.7 provides a good illustration of the above discussed process.

Example 2 $AAPL will be down today again but the down draft is slowing. By end of next week I think it's getting bought back.

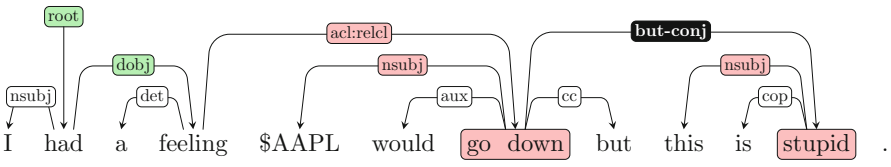

Fig. 5.7 Sentiment score propagates via the dependency tree (Example 1)

[4] http://cloud.google.com/natural-language

[5] Accessed on 2017-12-16.

Similar to Example 1, a period is missing at the end of this message, which is common among informal communication on the web. In addition, the preprocessing step converts the ASCII-based encoding for an apostrophe to its correct form and segments the message because there are actually two sentences. The first sentence is "*Apple will be down today again but the down draft is slowing.*" and the second is "*By end of next week I think it's getting bought back.*". In our experiments, Google SA predicts the sentiment score of the first sentence to be a negative -0.20 and the second sentence as neutral. As a result, the overall sentiment would be negative. However, this is wrong because the first sentence is just an objective description of a phenomenon and the second sentence apparently advocates for a bullish mood. Many people would even agree that both the sentences are conveying positive sentiment.

Indeed, user self-labeling of this message is positive. Augmented sentic computing arguably predicts a wrong sentiment score for the first sentence; however, correctly labeling the second as negative produces a correct overall sentiment polarity. In the first sentence, the but-conjunction governs two parts: "*will be down today again*" and "*the down draft is slowing*." In this sentic pattern, the polarity will be consistent with the latter part "*the down draft is slowing*." Because concept "`down_draft`" is not included in our knowledge base, the expression has to carry the polarity score of concept "`down`": -0.31. Another fault is that the knowledge base is not sensitive to the domain context: though in the financial domain "slowing of down draft" is positive (for asset prices), in a general sense, the concept "`is_slowing`" is neutral. Consequently, the sentiment score -0.31 passes throughout the entire sentence and marks the first sentence as negative. This example illustrates the importance to attach domain-specific concepts to the knowledge base to enhance model performance, which is the topic of chap. 6.

On the other hand, thanks to the analysis of the second sentence, we successfully revise the overall sentiment. Because "`bought_back`" is in every sense a strongly positive concept, it has a sentiment score of 0.82. Meanwhile, the adjective modifier relation (amod) carries the sentiment score -0.56 of "`next`" to "`next_week`." Additionally, the noun modifier relation between "`end`" and "`week`" gives the expression "*by end of next week*" an inverted (slightly) positive score of 0.02 (see Fig. 5.8). Finally, the sentiment score of the entire second sentence is computed as $1 - (1 - 0.82)(1 - 0.02) = 0.82$. The message averages the two sentence out: $(0.82 + (-0.31))/2 = 0.26$.

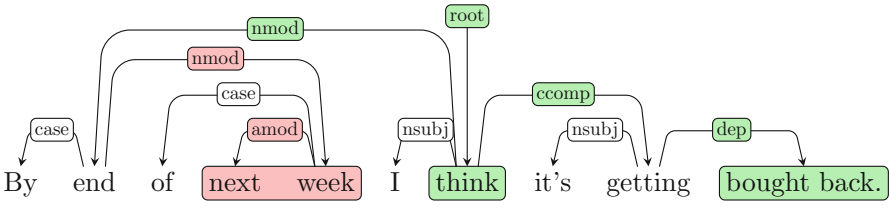

Fig. 5.8 Sentiment score propagates via the dependency tree (Example 2)

Example 3 $AAPL moment of silence for the 180 call gamblers. lol.[6]

This message raises the problem of Internet microtext: according to some statistics from large English corpora, we have to guess that "lol" may probably be an acronym for "laughing out loud." The speaker in a context tries to point out that recent price movements show that Apple's stock price would not reach 180, so the optimists stop advocating for their opinions (moment of silence). Moreover, the speaker sarcastically derogates the optimists as "gamblers." To capture this subtleness is often regarded as a difficult NLP task; Google SA predicts a positive sentiment score of 0.30 for this message. However, the user labeling is negative, and like for most of machine learning-based methods, to debug why and where this error is made is nearly impossible.

Augmented sentic computing not only gives the correct polarity direction but also clearly shows how this polarity score is concluded. Firstly, our dependency parser shows that the phrase "`moment_of_silence`" has a noun modification relation. Since "`moment`" is neutral, it inherits the sentiment of "`silence`" as 0.11. This sentiment is inverted for the whole sentence because of the case mark "for," which makes the overall sentiment direction depending on its latter part "*the 180 call gamblers*." Secondly, the concept "`gambler`" has a negative score of -0.74 in SenticNet, so the sentic pattern triggers a more intense negativity for the high-level multi-word expression
"`moment_of_silence_for_gamblers`" as $-\sqrt{|-0.74|} = -0.86$.

Let us consider the textual data stream from social media, which consists of messages anchored to different timestamp. We can count the daily positive and negative messages and compute the average sentiment score for any specific asset if we have a filtered data stream for this asset and apply augmented sentic computing to each message. Therefore, we define a sentiment variable s_t to represent the multidimensional sentiment time series for an asset A, which is at least a quadruple after quantization on a discrete-time axis, that is:

$$s_t(A) = (s_t^I(+), s_t^I(-), s_t^V(+), s_t^V(-)). \tag{5.16}$$

where $s_t^I(+)$ is the average *intensity* metric for all the positive messages in time period t; $s_t^I(-)$ is the average *intensity* metric for all the negative messages in time period t. Similarly, $s_t^V(+)$ is the aggregated count *volume* of positive messages regarding asset A and $s_t^V(-)$ count of negative messages [192]. While messages arrive continuously with a timestamp T_i, we do sentiment quantization on a daily basis to facilitate overnight trading strategy development. Specifically, to allow time for positions adjustment, we aggregate market sentiment from the previous trading day to 1 hour before market closure. Take the NYSE market as an example; messages posted from previous day 3:00 p.m. to current day 3:00 p.m. are aggregated for portfolio rebalancing during 3:00 p.m. to 4:00 p.m. local

[6]We have to guess that "lol" may probably be an acronym for "laughing out loud." [Or "lots of love", see https://www.computerhope.com/jargon///////l/lol.htm].

Algorithm 5.2: Sentiment time series construction

Data: message stream of a specific asset $\{m_i, T_i\}$
Result: sentiment time series $s_t(A)$

for $i = 1, 2, \ldots$ **do**
  if $T_i < t$ **then**
    $C(m_i) \leftarrow$ parse concepts from m_i;
    if $C(m_i) \bigcup KB \neq \emptyset$ **then**
      $s(m_i) \leftarrow$augmented sentic computing m_i;
    else
      $s(m_i) \leftarrow$MLP(m_i, Θ);
    end
    if $s(m_i) > 0$ **then**
      $s_t^I(+) \leftarrow \frac{n-1}{n} s_t^I(+) + \frac{1}{n} s(m_i)$;
      $s_t^V(+) \leftarrow s_t^V(+) + 1$;
    else if $s(m_i) < 0$ **then**
      $s_t^I(-) \leftarrow \frac{n-1}{n} s_t^I(-) + \frac{1}{n} s(m_i)$;
      $s_t^V(-) \leftarrow s_t^V(-) + 1$;
    $n \leftarrow n + 1$;
  else
    $t \leftarrow t + 1$;
    $[s_t(A), n] \leftarrow \mathbf{0}$;
  end
end
return $s_t(A) \leftarrow (s_t^I(+), s_t^I(-), s_t^V(+), s_t^V(-))$;

time. Algorithm 5.2 [192] provides a more detailed description of the construction process for sentiment time series. In the following section of data description, we demonstrate that this constructed time series can serve as a useful prior for stock market prediction and portfolio management even if the information is obtained from a public domain. We also cross validated the constructed sentiment time series and discovered that the time series calculated with augmented sentic computing exhibits similar patterns to users' self-labeling and some commercial tools, e.g., market sentiment product by PsychSignal.[7]

5.4 Data Description

The datasets we collected include messages from StockTwits,[8] which is "a popular social network for investors and traders to share financial information as well as their opinions" [189]. The investigation spans a time period of 3 months from 2017-08-14 to 2017-11-16 due to data availability. To construct a virtual portfolio of five stocks similar to that in Chap. 4, the dataset contains messages for five major stocks. In descending order of popularity, there are 38,414 messages for

[7] http://psychsignal.com
[8] http://stocktwits.com

Table 5.1 Confusion matrix between user labeling and sentic computing results

		Sentic computing		
		Positive	Negative	Total
User labeling	Positive	7234	3748	10,982
	Negative	2097	1445	3542
	Total	9331	5193	14,524

Apple Inc., 4298 messages for Goldman Sachs, 2847 messages for Starbucks, 2157 messages for Pfizer, and 1094 messages for Newmont Mining. Meanwhile, the sentiment time series from PsychSignal of the same period investigated contains 27,268 messages for Apple, 2298 messages for Goldman Sachs, 1844 messages for Starbucks, 826 messages for Pfizer, and 276 messages for Newmont Mining. We subsequently confirm that our raw dataset actually has a larger coverage than a common commercial product, though not all the messages would carry sentiment and be accurately labeled. After applying augmented sentic computing, we compare the obtained sentiment analysis results and user labels in the confusion matrices. Table 5.1 is an example for Apple: from the 38,414 messages that mentioned Apple only 14,524 messages are user labeled and thus eligible for comparison. The accuracy of polarity detection by sentic computing can be easily calculated as 59.8% from Table 5.1. This accuracy can be considered prestigious because the raw data is noisy and only a general domain sentiment knowledge base (SenticNet) is used. Another issue worth mentioning is that user labeling *cannot* be fully understood as a ground truth, but only as a reference. Some message will lose and change the sentiment as contexts get lost. Therefore, "agreement ratio" may be a more precise term to describe this metric. It is common (not only for Apple) that only a small portion (usually less than 20%) of users will label their messages, but we cannot assume they are an unbiased sample for the population of sentiment expressed.

The second dataset we collected is not in the format of natural language but a processed "trader mood index" from a third-party commercial product, i.e., PsychSignal. The data investigates a period of around 8 years (2800 days from 2009-10-05 to 2017-06-04). The way they calculate sentiment time series is described as first filtering multiple sources, including Stocktwits, Twitter data, and others by cashtags[9] of the query ticker, and applying NLP techniques to compute the sentiment intensity scores.[10] Figure 5.9 segments a time period of 90 days (2017-03-04 to 2017-06-04) and visualizes the public mood data stream from PsychSignal on Apple Inc. The illustration mainly includes four dimensions: volume of daily tweets (blue, left); average sentiment intensity (red, left); net sentiment polarity (red, right); and daily returns (black, right). We can observe a periodic weekly cycle of message volume in Fig. 5.9: because on weekends and market closure days many fewer messages will be posted.

[9]Cashtags are stock tickers prefixed with a dollar sign that are widely used for sharing financial information in social media [74].

[10]The detailed techniques of how their sentiment analysis engine works is not disclosed.

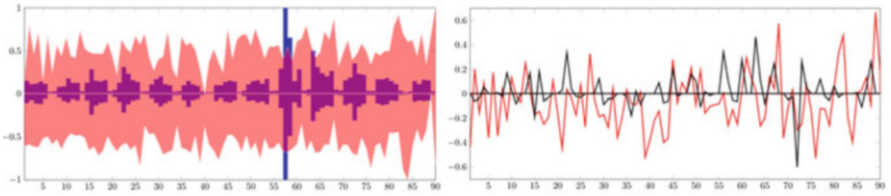

Fig. 5.9 PsychSignal sentiment stream (cashtag "AAPL", normalized)

Although the information source is not from the same one, the visualization of positive and negative message counts time series clearly exhibit a consistency between PsychSignal and StockTwits processed with augmented sentic computing (Fig. 5.10). Our method can also produce sentiment intensity data; however, comparison is not made because the corresponding part is missing from the PsychSignal data.

The correlations of any two sentiment time series are further calculated as:

$$\text{Correlation}(\mathbb{S}_1, \mathbb{S}_2) = \frac{\mathbb{E}((\mathbb{S}_1 - \bar{\mathbb{S}_1})(\mathbb{S}_2 - \bar{\mathbb{S}_2}))}{\sigma_{\mathbb{S}_1}\sigma_{\mathbb{S}_2}}. \tag{5.17}$$

Table 5.2 provides the correlations among the message sentiment time series from three different sources: user labeling, sentic computing, and PsychSignal. We find out that all the correlations are positive and significant. Moreover, we obtain additional data to prepare for the intelligent portfolio management strategies. Specifically, the daily closing stock prices and trading volumes are from the Quandl API[11]; the market capitalization size data are from Yahoo! Finance. For missing values, such as the closing prices on weekends and public holidays, we fill the gap with the closest historical data, so that allocation strategies can be learned on a daily basis.

5.5 Experiments

Our asset allocation strategies can be assessed in two stages: the first one on the helpfulness and quality of "market sentiment views" and the second one on the capability of different implementations of the approximator $\mathbb{F}$. For the two stages, we construct a representative portfolio and implement two experiments, respectively. Similar to that in Chap. 4, our portfolio comprises five stocks: Apple Inc (`AAPL`), Goldman Sachs Group Inc (`GS`), Pfizer Inc (`PFE`), Newmont Mining Corp (`NEM`), and Starbucks Corp (`SBUX`). The motivation for such a portfolio is that

[11] See Sect. 4.3.

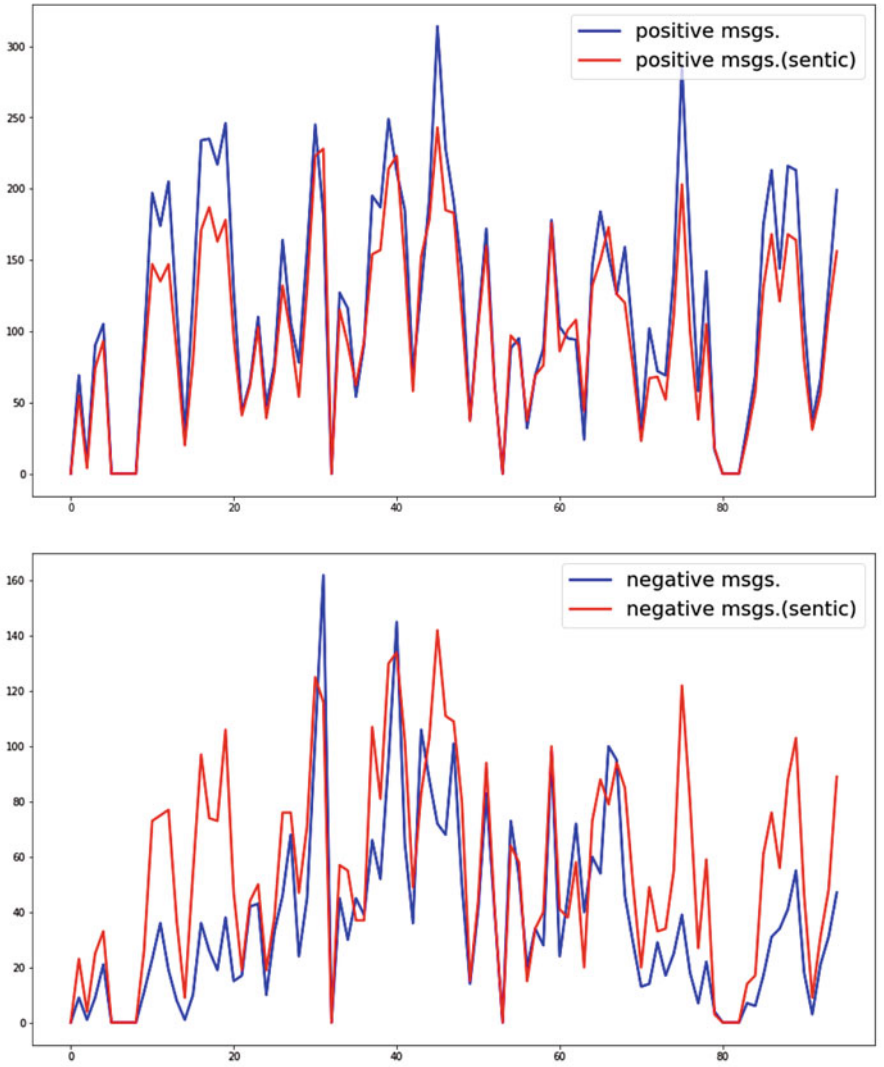

Fig. 5.10 The time series of positive and negative message counts from two sources

Table 5.2 Correlation of message sentiment time series [189]

	Positive messages	Negative messages
User-Sentic	+0.964	+0.795
User-Psych	+0.185	+0.449
Sentic-Psych	+0.276	+0.282

in the practices of asset management, a rule of thumb is to diversify the investments, while these five stocks cover the two major US markets (NYSE and NASDAQ) and also diversified industries, such as technology, financial services, healthcare, consumer discretionary, etc. Moreover, we include both high-tech companies, i.e., Apple, and an industrial company, i.e., Newmont, because their message volume and post frequencies differ a lot. On social media the traditional industries receive less discussion while high-tech industries get more attention. The stock prices per share are normalized to the current numbers if there is any split history. We do not consider dividends because one cannot expect such activities and once done, the adjustment in prices will not affect allocation decisions in future. In consistency with previous settings, we allow no short selling, taxes, or transaction fees in the simulations. We also assume continuous and infinitely divisible amounts of investments starting from 10,000 dollars in sum are available.

We use various metrics, including RMSE, CAGR,[12] Sharpe ratio, Sortino ratio, and MDD to evaluate the performance of virtual portfolios. RMSE is probably the most common metric for approximation/regression problems. It calculates the standard deviation of the model predictive errors and is frequently used in study fields of engineering. RMSE is considered a good metric when the data are normally distributed and contains few outliers. We take our realized portfolio weights as model outputs and the optimal weights as true values to calculate RMSE, so that it measures the difference between the realized portfolio-weighting strategy and the a posteriori optimum:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|w_i - \hat{w}_i\|^2} \tag{5.18}$$

The Sharpe ratio is a risk-adjusted return measure that helps investors understand the expected return with per unit of risk. We choose EW or VW as the base portfolio, so that its Sharpe ratio will be 1.00:

$$\text{Sharpe ratio} = \frac{\mathbb{E}(R_{pfl}/R_{\text{base}})}{\sigma(R_{pfl})/\sigma(R_{\text{base}})}. \tag{5.19}$$

It is worth mentioning that the absolute numbers of risk are not comparable if calculated from different frequencies or timespans. We use the standard deviation of daily returns as a measure of risk. Some argues that the good part should be distinguished and considered separately from bad risk. The solution is to use the standard deviation of downside returns only to estimate risk. This single-side measure is called the Sortino ratio [158].

MDD (maximum drawdown) measures the maximum possible percentage loss of an investor:

[12] See equation 4.17.

$$\text{MDD} = \max_{0<t<\tau} \left\{ \frac{C_t - C_\tau}{C_t} \right\}. \tag{5.20}$$

MDD is usually considered as an indicator of tolerance to psychological pressures. Portfolio management strategies with large MDDs tend to give rise to panic and impatience among investors and increase the risk of capital withdrawal.

5.5.1 Simulation: Effectiveness of Market Views

The results of intelligent asset allocation with sentiment are benchmarked with two portfolio construction strategies:

1. The value-weighted portfolio (VW): we reinvest the capital daily according to the percentage share of each stock's market capitalization. This is a slightly improved version of the equal-weighted portfolio (EW, see Sect. 4.4.2) with the rationale that bad-performing stocks shrink as other investors quit their positions. In VW the portfolio performance will be the weighted average of each individual stock's performance. This strategy is also a fundamental yet tough-to-beat baseline.
2. The neural trading portfolio (NT): we skip the formation of market sentiment views and directly train a neural network with the same full input including sentiment information. The target outputs are daily optimal portfolio weights. This benchmark serves as an ablation analysis not for the sentiment information, but for the market view construction. In this circumstance, we are not able to get insights on why the portfolio weights should have such values. Thus the NT strategy is a black-box strategy.

We implement several portfolio settings as follows:

1. No views portfolio ($\Omega_\varnothing$): since the expected return distribution and correlations will in this case be the same as the estimators in a classic Markowitz's mean-variance portfolio, this model degenerates to the normal Markowitz's settings.
2. Random views portfolio (Ω_r): the parameters for market views are randomly imposed.
3. Standard views portfolio (Ω_0): the confidence of views are formed using the construction of Black-Litterman model, while the expected return vector is estimated either with or without the sentiment time series ($\mathbb{S}$).

The trading simulation performances with different experimental settings are demonstrated in Fig. 5.11, where the x-axis denotes the index of trading days and the y-axis denotes the amount of cumulative capital. Because we have long enough historical price records, we try two window sizes (90 and 180 days) to batch the data for our training processes. Another factor beside window size that would affect the model performance is the selection of an approximator neural network. Therefore, we experimented with both DENFIS and LSTM, and the results are compared in Fig. 5.11c, d, respectively, for better presentation.

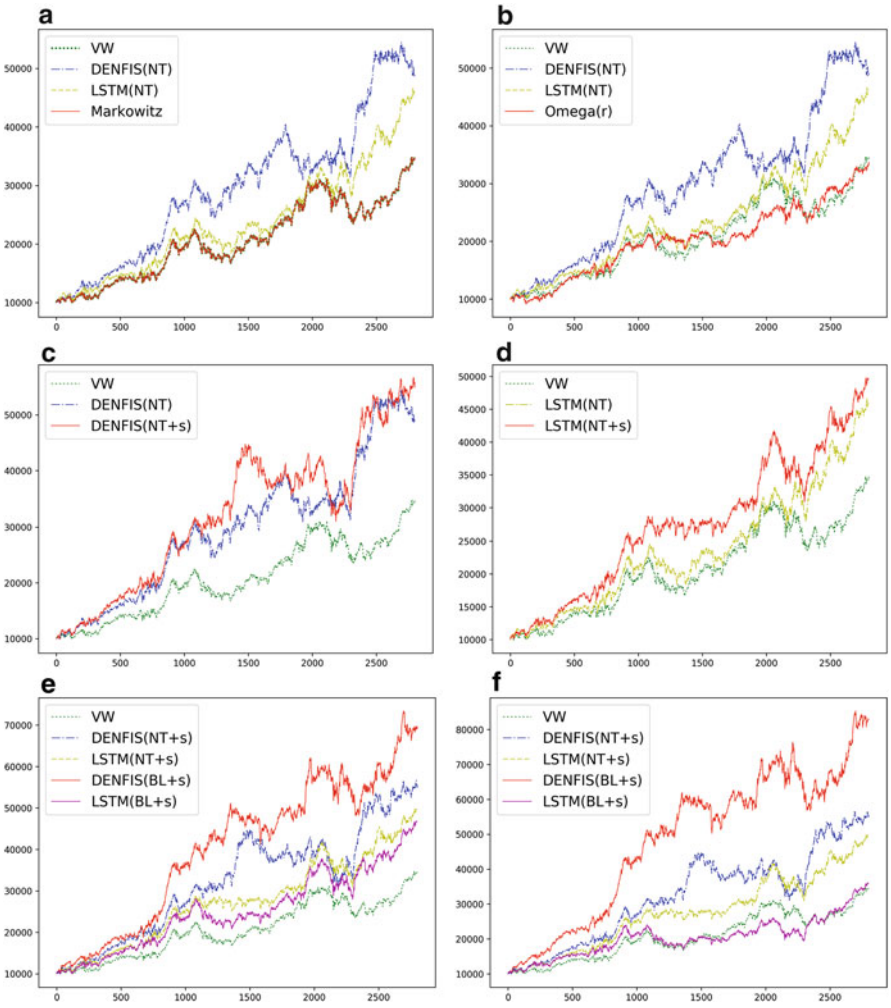

Fig. 5.11 Trading simulation performance with/without market sentiment views. (**a**) No views. (**b**) Random views. (**c**) DENFIS + sentiment. (**d**) LSTM + sentiment. (**e**) BL + sentiment, t = 90. (**f**) BL + sentiment, t = 180

To keep aligned with the previous study, we use a risk aversion coefficient as $\delta = 0.25$ and confidence level of CAPM as $\tau = 0.05$. These are common values reported from literature. By minimizing the global portfolio weight error during training, we empirically choose the activation range of the fuzzy membership function to be $d = 0.21$. The final network has 21 fuzzy rule nodes from the entire online training process of DENFIS. For the recurrent neural network-based approximator, we stack two LSTM layers and connect a densely connected layer at the end. Each LSTM layer has only 3 cells, but the densely connected layer has 50 neurons according

Table 5.3 Performance metrics for various view settings [189]

	RMSE	Sharpe ratio	MDD(%)	CAGR(%)
VW	0.8908	1.00	25.81	17.49
Markowitz90($\Omega_\varnothing$)	0.9062	1.00	25.81	17.51
Markowitz180($\Omega_\varnothing$)	0.8957	1.00	25.82	17.45
BL90(Ω_r)	0.9932	0.90	23.47	17.17
BL180(Ω_r)	0.9717	1.06	**20.59**	22.31
DENFIS(NT)	0.9140	2.94	29.84	23.09
DENFIS(NT + $\mathbb{S}$)	0.9237	**4.35**	23.07	25.16
DENFIS(BL90 + $\mathbb{S}$)	0.9424	1.52	24.44	28.69
DENFIS(BL180 + $\mathbb{S}$)	0.9490	1.58	24.19	**29.49**
LSTM(NT)	0.8726	1.38	25.68	22.10
LSTM(NT + $\mathbb{S}$)	0.8818	1.42	25.96	23.21
LSTM(BL90 + $\mathbb{S}$)	**0.8710**	1.34	25.90	22.33
LSTM(BL180 + $\mathbb{S}$)	0.8719	1.07	24.88	17.68

to the idea that it should be at least two or three times larger than the LSTM layer size. The objective of training is to minimize the mean squared error of vector Q as loss function. We discovered that the *rmsprop* [169] optimizer achieves the best performance. Fortunately, the training error in our experiments always converges quickly.

We provide quantitative metrics in Table 5.3 as a supplement to Fig. 5.11 for the purpose of assessing different methods. Acronyms in the table such as "BL90", e.g., denote market views of the Black-Litterman model with timespan = 90 days.

Some interesting observations can be spotted from Fig. 5.11 and Table 5.3. First, though the intuition is that the portfolio performance should be better if the actual portfolio weights are close to the optimal weights, closeness in what sense makes a difference. RMSE is a commonly used metric of closeness, but its relation to other three performance metrics is weak. This is because the relationship between weights and daily returns is *nonlinear*. Consequently, errors are not of the same importance in terms of maximizing the final portfolio return. A more severe problem caused by it is that the LSTM models seem to overfit as they are trained on the mean squared error of weights or expected return of views [130]. Therefore, it is not recommended to use any metrics outside the field of asset allocation to evaluate expected portfolio performances [189]. Arguments that used those metrics, such as directional accuracy of price change prediction, e.g., [19, 198], are not sound in the context of portfolio management.

Figure 5.11a shows that the behavior of a Markowitz portfolio has almost no difference to the market following strategy (VW). This also explains why the mean-variance approach is inefficacious in practice, as discovered by many previous studies. In real world, if the CAPM holds, the market portfolio (VW) would have already reflected the adjustments to risk premiums. The dynamics are described

from the opposite direction: because fewer market participants will invest on highly risky assets, their market capitalization share will be smaller as well.

Since the Markowitz model has no advantage over the value-weighted portfolio, a natural question to ask next is "will the Black-Litterman model do a better job?" Our experiments show that a better performance over the Markowitz portfolio is possible, whereas not guaranteed. The key issue is the quality of input views of the Black-Litterman model: "garbage in, garbage out" still holds under this circumstance. A baseline is given by the portfolio with random views (Ω_r). Obviously, the performance can be worse than market following (VW) in terms of both Sharpe ratio and CAGR. The lesson learned here is a clear understanding of the capability of using the Black-Litterman model. When the investor knows nothing he/she inevitably inputs pseudo-random views. By doing so he/she pretends to know something, but the outcome will be worse than to assume no views and follow the market.

In these comparisons, the DENFIS-based approximator usually performs better than LSTM-based models, achieving higher Sharpe ratio and CAGRs. Through analyses of predicted weights, we suspect that LSTM models adapt too fast to the incoming data. While financial time series are considered very noisy, the adaptation may not always be beneficial because it catches spurious signals. For DENFIS, the ECM mechanism controls model learning rates, thus providing stability to the memorized fuzzy rules. It is worth mentioning that regardless of the neural network implementations, blending of sentiments improves CAGRs for both DENFIS and LSTM. The timespan used to estimate correlation and volatility of assets seems not that critical—the difference between using 90 days and 180 days is trivial. It seems that a longer timespan fits the DENFIS-based models, while using a shorter timespan the LSTM-based models perform better. The Markowitz portfolio is less affected by timespan since sentiment information cannot be used.

5.5.2 Simulation: Effectiveness of ECM-LSTM

The results of implementing ECM-LSTM with different sentiment sources are benchmarked with not only the LSTM model but also three other portfolio construction strategies as follows. As previously justified, the ECM-LSTM implementation is a machine learning model. To allow comparison with statistical models, we introduce the ARIMA portfolio and the HW portfolio of autoregressive nature.

1. The equal-weighted portfolio (EW): we hold equal weights (20%) for the five stocks in our portfolio throughout the examined period. This setting is the same as elaborated in Sect. 4.4.2.
2. The ARIMA portfolio (ARIMA): we re-invest the capital daily according to the one-step-forward price forecasts. Obviously, the holding weights will be a one-hot vector. The price forecasts are produced by an ARIMA(p, d, q) process, where the parameters p, d, q are estimated from the past time series

data following the Box-Jenkins method. First, we increase d from zero until the differenced time series is stationary according to the augmented DickeyFuller (ADF) statistic. Next, we set the upper bound of p and q as the orders of the last significant partial autocorrelation and autocorrelation. Finally, we select pair (p, q) that has the minimum Akaike information criterion (AIC). After these steps we identified an ARIMA(0, 1, 2) model for `PFE` and ARIMA(0, 1, 0) models for other stocks, which means that the price series are not much different from random walk.

3. The Holt-Winters portfolio (HW): we re-invests daily according to the one-step-forward price forecasts as well. However, the forecasts are produced by a Holt-Winters additive smoothing method with time-varying parameters. The model HW($\hat{\alpha}_t,\ \hat{\beta}_t,\ \hat{\gamma}_t$) is specified at each time point t by minimizing RMSE of simulated time series in a sliding window $(t - k,\ t)$.

Although the ARIMA and HW portfolios do not leverage any external information, that is, all the model parameters are estimated from past observations, they are considered to be among the most effective forecasting techniques across different tasks such as for crude oil prices and macroeconomic indices when no useful prior is available [106]. In contrast to the ARIMA and HW portfolios, we also construct portfolios that take into account sentiment time series from different sources using the Black-Litterman model. The customized LSTM architecture contains one layer of 64 LSTM cells followed by another densely connected layer of the size of the number of assets. The LSTM layer has 20% dropout ratio to avoid over-fitting. The simulated trading results are shown in Fig. 5.12 and Table 5.4 presents the corresponding metrics.

Figure 5.12 demonstrates that despite the facts that sentiment information is from different sources and implementation details of approximator $\mathbb{F}$ for market views also differ, the curve patterns of accumulated return are similar. These patterns can be regarded as systematic movements to the time period and portfolio. Adding the ECM mechanism on top of LSTM effectively mitigates crashes, as in two of three ECM-LSTM-based portfolios, the 2017-09-15 to 2017-09-25 period witnesses a correction to the capital loss (see Fig. 5.12).

The top 3 of each metric in Table 5.4 are marked in bold. In terms of Sharpe ratio and Sortino ratio, surprisingly, EW is the best strategy in our experiments. EW is also a very stable strategy in a sense that it has the minimum MDD. ARIMA and HW are more volatile than EW. This is probably because, after forecasting of next-day prices, the whole capital is invested to the only winning asset; thus, the risk of this prediction error is not well diversified. CAGRs of the ARIMA and HW portfolios cannot compete with the EW in the experiments as well, resulting in very small Sharpe ratio and Sortino ratio for the ARIMA and HW portfolios. Therefore, we showcase that though these two models perform well in general tasks, they are not preferred for constructing portfolios, at least in a manner of full position rebalancing based on price forecasts.

All the portfolios that have considered market sentiment, no matter using what information source and implementation details, achieve higher CAGRs than the

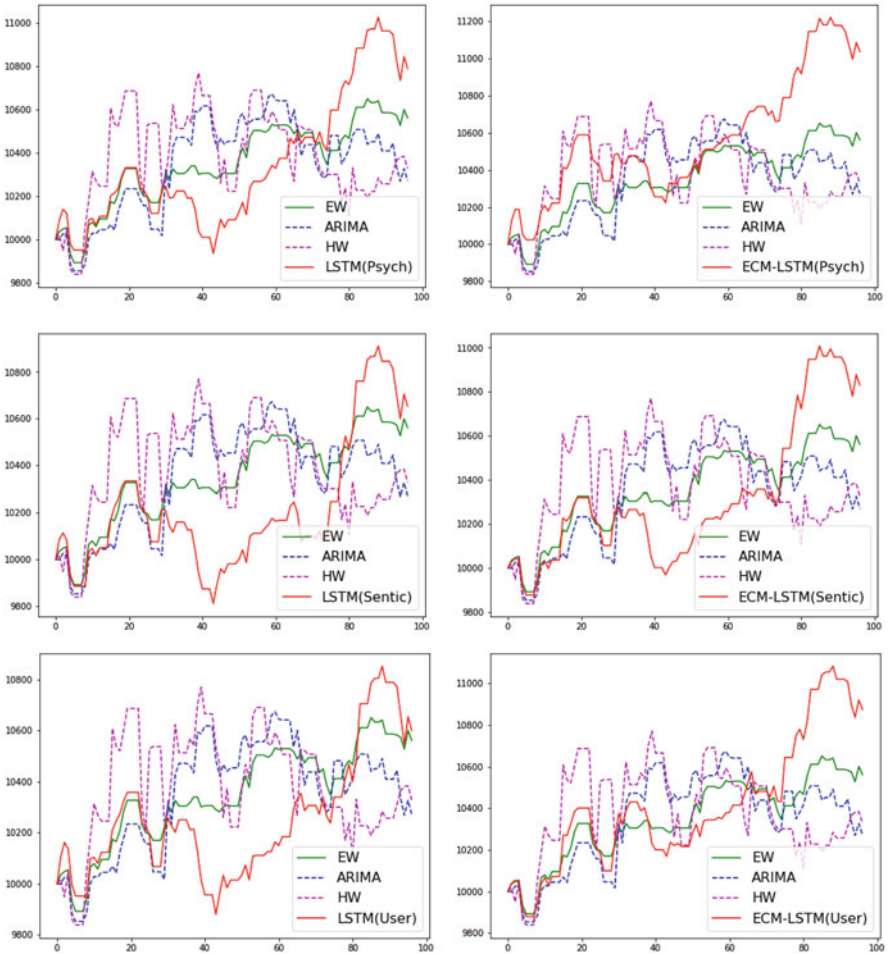

Fig. 5.12 Trading simulation performance with different sentiment sources.

Table 5.4 Performance metrics for different sentiment sources [189]

	Sharpe ratio	Sortino ratio	MDD(%)	CAGR(%)
EW	**1.00**	**1.00**	**1.76**	23.07
ARIMA	0.56	0.61	3.79	10.72
HW	0.34	0.36	6.16	13.03
LSTM(Psych)	0.71	0.79	3.84	33.52
LSTM(Sentic)	0.61	0.68	5.05	27.21
LSTM(User)	0.64	0.68	4.61	24.82
ECM-LSTM(Psych)	**0.74**	**0.82**	3.45	**45.51**
ECM-LSTM(Sentic)	0.66	0.73	**2.89**	**35.45**
ECM-LSTM(User)	**0.71**	**0.87**	**3.40**	**37.53**

three benchmark strategies (EW, ARIMA, and HW) discussed before. This confirms the importance of leveraging market sentiment. Moreover, the improvement of introducing an ECM mechanism is confirmed by the fact that in terms of all these metrics, the ECM-LSTM portfolios systematically outperform their LSTM counterparts using the same source of sentiment information.

In our experiments, Sortino ratios are slightly greater than Sharpe ratios according to Table 5.4. This is because the market trend in the time period we examined is going bullish. Therefore, the optimistic measure of only downside risk with Sortino ratios is smaller. Nevertheless, in terms of the absolute value all the strategies have Sharpe ratios and Sortino ratios less than 1.00. This phenomenon may be due to the fact that in those "efficient markets," obtaining cheap premium return is really difficult. In pursuit of higher CAGRs, investors inevitably take a greater unit risk because the low-hanging fruit (risk-free interest) is already picked.

Finally, we discuss the differences in portfolio performances among using different sentiment information sources. The quality of sentiment information is intuitively important; however, in our experiments no evidence suggests a clear difference. PsychSignal seems to be the source that provides the most accurate sentiment time series data because (1) it has the largest message volume and (2) the portfolio leveraging this source performs the best. However, using just the user labeled message counts sometimes also achieved a balanced and advantageous result. A more detailed comparison of different sentiment information sources will not be a topic covered in this book.

5.6 Summary

Market sentiment appears to be increasingly attractive in the computational intelligence and econometrics communities. However, when leveraging such information, the problem is often formulated as to assist price forecasting rather than to allocate and manage among a pool of assets. The latter task formulation is apparently more practical and produces new challenges. This chapter pioneers computing sentiment-induced market views from social media data stream and integrating it into the state-of-the-art asset allocation method, namely, the Black-Litterman model. Cross-validation and intensive experiments suggest that the sentiment time series can be obtained using augmented sentic computing—a concept-level sentiment analysis approach. When applied to asset management, the efficacy of this sentiment time series is comparable to some established commercial tools.

The unique advantage augmented sentic computing has over other candidate sentiment analysis methods is its transparency and good interpretability. As a result, it has a great potential for broader NLP-based financial applications because financial service features strong regulations and explanability to build trust. With the assistance of the formalization and market views computation, some insights are brought to the daily asset reallocation decisions. We tell a story of the rationale behind portfolio rebalancing decisions using 2017-06-01 as an example.

"On June 1st 2017, we observe 164 positive opinions of polarity +1.90, 58 negative opinions of polarity −1.77 on AAPL stock; 54 positive opinions of polarity +1.77, 37 negative opinions of polarity −1.53 on GS stock; 5 positive opinions of polarity +2.46, 1 negative opinion of polarity −1.33 on PFE stock; no opinion on NEM stock; and 9 positive opinions of polarity +1.76, 5 negative opinions of polarity −2.00 on SBUX stock. Given the historical prices and trading volumes of the stocks, we have 6.29% confidence that AAPL will outperform the market by −70.11%; 23.50% confidence that GS will outperform the market by 263.28%; 0.11% confidence that PFE will outperform the market by −0.50%; 1.21% confidence that SBUX will outperform the market by 4.57%. Since our current portfolio invests 21.56% on AAPL, 25.97% on GS, 29.43% on PFE, and 23.04% on SBUX, by June 2nd 2017, we should withdraw all the investment on AAPL, 2.76% of the investment on GS, 81.58% of the investment on PFE, and 30.77% of the investment on SBUX, and re-invest them onto NEM."

For each day, this template-based story can be disclosed to the investors to provide them confidence and build trust. More discussions can be found with regard to robo-advisory in Chap. 7.

Chapter 6
Storage and Update of Knowledge

Learning without thought is labor lost; thought without learning is perilous.

—Confucius

Abstract Experience in developing large knowledge-based AI projects suggests a progressive approach: the system needs maintenance to keep pace with demands and accumulation of commonsense knowledge to prevent having to start all over again. Financial asset management is no exception. The balance between leveraging the current knowledge base and adding to it is analogous to the learning and thought relation described by Confucius. In the previous chapter, sentic computing is actively "thinking" with the knowledge base, however, not learning anything. An example also shows the problem of unable to retrieve domain-specific concepts from the knowledge base. In this chapter, discussions on the forms of storing semantic and sentiment knowledge are presented. A special effort on adding and updating polarity scores of words with high-level supervision is made. The same idea can be extended to other application domains as well as the curation of concepts or events.

Keywords Knowledge representation · Ontology engineering · Financial sentiment lexicon · Polarity score · Domain adaptation · Heuristic search

6.1 Storing Semantic and Sentiment Knowledge

In Sect. 5.3.1 we introduced how semantics and sentics are stored in SenticNet. As an open-domain resource, SenticNet only associates related concepts without specifying the relation type. In the finance domain, at least two kinds of relations are important, namely, is-a relation and causal relation. We conceive an ontology of semantic knowledge, which is enriched by three tools: a concept parser, a taxonomy parser, and a causal parser. The concept parser identifies concepts in a sentence by two shreds of evidence: the n-gram probability and POS compatibility. Take the example "*Because of the yen's appreciation, the Japanese economy deteriorated*"

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4_6

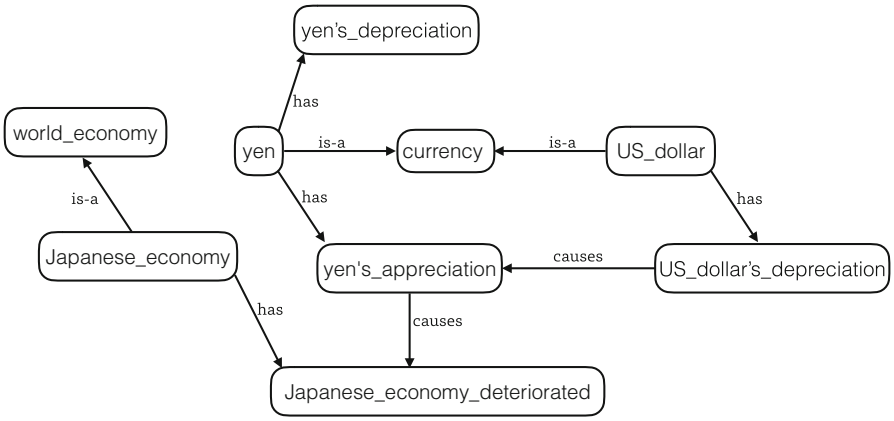

Fig. 6.1 Visualization of the ontology of semantic knowledge

from [145]. A concept "`yen's_appreciation`" is extracted because this bi-gram has a high frequency from the financial text corpus and is a noun phrase (noun+noun).[1] In contrast, though "of the" is an even more frequent bi-gram, preposition-determiner is not a valid concept type.

Taxonomy parser and causal parser extract "is-a" and "causes" relations. In the abovementioned case, a causal relation:

$$\texttt{yen's\_appreciation} \Longrightarrow \texttt{economy\_deteriorated}$$

is extracted. As described by [145], this process is enabled by syntactic features and keyword lists.

Once the relation is extracted from texts, we will check whether there exists a conflict in the knowledge base. If the confidence score is higher than that in the knowledge base, we will insert this relation and resolve the conflict. A final ontology may look like Fig. 6.1.

6.1.1 From Sentiment Lexicon to Sentiment Knowledge Base

Similarly, sentiment knowledge can be stored in a graph-based structure. The relations, however, are simpler, because theories of emotions conclude a small number of categories for sentiment words and concepts. Therefore, the data structures used for storing and manipulating sparse matrices can apply. A simple sentiment lexicon may only contain a list of positive words and a list of negative words, e.g., as in

[1] Other concept types includes adjective + noun, verb + noun, etc.

Table 6.1 Positive and negative word lists of Opinion Lexicon

Positive words	Negative words
a+	Two-faced
Abound	Two-faces
Abounds	Abnormal
Abundance	Abolish
Abundant	Abominable
Accessable	Abominably
Accessible	Abominate
Acclaim	Abomination
...	...

```
<rdf:RDF xmlns:rdf="http://w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="http://sentic.net/api/en/concept/meet_friend">
    <rdf:type rdf:resource="http://sentic.net/api/concept"/>
    <text xmlns="http://sentic.net">meet friend</text>
    <semantics xmlns="http://sentic.net">
      <concept xmlns="http://sentic.net" rdf:resource="http://sentic.net/api/en/concept/meet_person"/>
      <concept xmlns="http://sentic.net" rdf:resource="http://sentic.net/api/en/concept/chit_chat"/>
      <concept xmlns="http://sentic.net" rdf:resource="http://sentic.net/api/en/concept/make_friend"/>
      <concept xmlns="http://sentic.net" rdf:resource="http://sentic.net/api/en/concept/meet_girl"/>
      <concept xmlns="http://sentic.net" rdf:resource="http://sentic.net/api/en/concept/socialize"/>
    </semantics>
    <sentics xmlns="http://sentic.net">
      <pleasantness xmlns="http://sentic.net" rdf:datatype="http://w3.org/2001/XMLSchema#float">0.048</pleasantness>
      <attention xmlns="http://sentic.net" rdf:datatype="http://w3.org/2001/XMLSchema#float">-0.08</attention>
      <sensitivity xmlns="http://sentic.net" rdf:datatype="http://w3.org/2001/XMLSchema#float">0.036</sensitivity>
      <aptitude xmlns="http://sentic.net" rdf:datatype="http://w3.org/2001/XMLSchema#float">0</aptitude>
    </sentics>
    <moodtags xmlns="http://sentic.net">
      <concept xmlns="http://sentic.net" rdf:resource="http://sentic.net/api/en/concept/joy"/>
      <concept xmlns="http://sentic.net" rdf:resource="http://sentic.net/api/en/concept/surprise"/>
    </moodtags>
    <polarity xmlns="http://sentic.net">
      <value xmlns="http://sentic.net">positive</value>
      <intensity xmlns="http://sentic.net" rdf:datatype="http://w3.org/2001/XMLSchema#float">0.031</intensity>
    </polarity>
  </rdf:Description>
</rdf:RDF>
```

Fig. 6.2 Entry for concept "`meet_friend`" in SenticNet

Table 6.1 [76]. More detailed lexicon may have a polarity score for each word entry to denote the intensity of the sentiment, e.g., SentiWordNet [5].

One problem of having only one polarity score for each word is polysemy, i.e., different sense of a word may have different sometimes even opposite polarity. Many concept polarities cannot resort to the component words as well. For example, "`pain_killer`" is a positive concept, but "pain" and "killer" are negative words. This problem can be solved in either a symbolic or sub-symbolic way. Ren et al. [141] suggested that polysemy can be modeled with multiple prototype word embeddings. We can also, to some extent fix the semantic role of words by allowing concepts into the knowledge base. SenticNet [28], for instance, is such a sentiment knowledge base that further gives associated concepts, multiple dimensions of the sentiment, and mood tags. Figure 6.2 provides an example of concept entry in SenticNet using RDF Schema.

6.2 Cognitive-Inspired Domain Sentiment Adaptation

Consider a simplified case, where sentiment lexicon contains only words and their polarity scores. We study possible approaches to updating words and adapting existing polarity scores to the finance domain. This serves as an important infrastructure for accurate financial sentiment analysis and sentiment time series construction, which further supports the asset allocation models. The following content is adapted and extended from [194].

The identification of specified domain and the adaptation to it are two of the main issues in sentiment analysis [16]. While using the lexicons of the common domain, the performance of domain-specific sentiment is dropped along with other domains. The finance domain has its specific terminologies and languages and is characterized by its sub-languages and jargons. Due to these reasons, it is suboptimal in directly using the lexical resources l [36].

Many studies have discussed domain adaptation for the sentiment lexicon to deal with challenges [164]. It's a matter of fact that word-level supervision is difficult to access in this situation because latent information is expressed in the form of word polarities. Rating websites and social media are used for language resources of sentiment analysis, and user provided the supervisions. Hence supervision is essential for high-level accuracy, e.g., expression-level, sentence-level, and document level. Choi and Cardie [36] used integer linear programming for formulating the lexicon programming tasks to evaluate the word-to-word relation and also a word-to-expression relation. Many other relations include n-grams, POS, term frequency [50, 118], TF-IDF and its variants, (positive) PMI [70, 110, 174], etc. Apart from the real words, many other words that do not belong to any vocabulary can be added in sentiment lexicons [180], and a constructed graph can be drawn to check their polarity scores for propagation. A hand-crafted thesaurus can be formulated for non-vocabulary words [178]. These graphs can be designed based on corpus statistics, which contains a similarity matrix in the context [165] or word embedding [167]. Ofek et al. formulated the acyclic graph from the statistical co-occurrence information for the purpose to enrich a concept-level sentiment lexicon [126]. However, in their methods, during the learning phase, the polarity scores are not exposed to the original sentiment lexicon. Therefore, we propose a novel cognitive-inspired approach during the learning phase that expects to change the polarity score value. Different metacognition processes are part of this approach when the domain of any new language is exposed: based on the information of word polarities, the agent made presumptions [131]; his lexical information cannot be changed before the identification of conflict; therefore he would try to implement different information and try to locate a word; the information is then subjected to future occasions for approval. Figure 6.3 depicts the fundamental idea of this approach.

The negative record has been predicted for the first score due to words *dump* and *loss*, while the record is positive for the user. It has been observed that neutral word *small* showed a positive polarity. While the second record against revised the

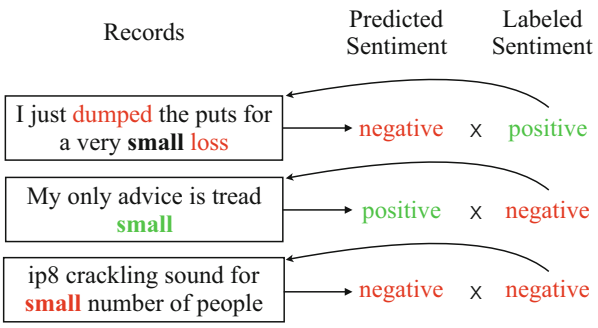

Fig. 6.3 Illustration of the polarity score adaptation process of word *small* [194]

information. On the contrary, the third record again showed the negative polarity *small*. A word *cracking* carries negative sentiment, and polarity values remain the same in this situation. Different experiments have been performed for the demonstration of this approach.

6.3 Methodology

Firstly, polarity value has been denoted for the word x by $\gamma(x)$. The representation of the sentiment lexicon $\mathscr{L}^D(\mathbf{x} : \gamma(\mathbf{x}))$ as a starting point, the size for the vocabulary $\mathscr{L}$ is D. We have set the different training sets including N records T_i, where $1 \leq i \leq N$. While the corresponding sentiment label for T_i is y_i. Hence, we can use the different algorithm for classification of sentiment and algorithm like $\mathbb{F}_{\mathbf{T}}(T, \mathscr{L})$ that outputs a prediction label y for input record T. The specific choice of $\mathbb{F}(\cdot)$ is independent of the following steps.

6.3.1 Vectorization of Sentiment Features

Apart from sentiment lexicons and polarity value, different other features have been proposed for the classification of sentiment. These features are based on term frequencies, POS tags, negators, syntactic features, and more [107]. All of these features can be used to train the classifier, and, in fact, having sentiment lexicons does not take much advantage in training. However, this study aims to demonstrate a way to adapt the sentiment lexicon rather than to train the best classifier. To this end, we only make use of the relevant sentiment information for training.

Each record T is represented with a D-dimensional vector, $vector(T)$, where the dimension indices of the vector indicate the unique location of the word in $\mathscr{L}$.

That is, for $x = vector(T)_i$, if x is in $\mathscr{L}$, the polarity score of $x \in [-1, 1]$ will be assigned to $vector(T)_i$:

$$vector(T)_i = \begin{cases} 0 & \text{if } x \notin \mathscr{L} \\ \gamma(x) & \text{if } x \in \mathscr{L} \end{cases} \tag{6.1}$$

In this study, a binary label $y \in \{+, -\}$ is used to denote the positive or negative result for the sentiment classification information.

6.3.2 Exploration-Exploitation

The algorithm is used after the training process, for each record T_i, and for the predicted sentiment, the label is used as $\hat{y}_i = \mathbb{F}_{\mathbf{T}}(T_i, \mathscr{L})$ for checking it against the ground realities. If in case of the start algorithm, the symbolic representation used as $\hat{y}_i \neq y_i$, then the error can be corrected by using better symbolic representation to correct the error. However, such errors can be linked with any word $vector(T_i)_j$ in $vector(T_i)$. Unrealistic computer power is required for word polarity combinations. Hence, we are unable to achieve the balance between correct polarity score and predicted label words for polarity score. There are different psycholinguistic theories that discussed the fact that human behavior is associated with varying levels of sentiments and activated words of sentiments [27, 92]. Naturally, activated words, i.e., where $vector(T_i)_j \neq 0$, are explored in descending order of the absolute value of their polarity scores.

For every single word $x \in \{vector(T_i) \cap \mathscr{L}\}$, the polarity score assignment is performed by an algorithm according to the following rule:

$$\gamma(x)' = \gamma(x) + \zeta \tag{6.2}$$

where ζ is a number that is evolved from uniform sharing, and it is a random float number. The decision would be taken by algorithm that adaption of the new polarity score $\gamma(x)'$ must be done before searching for some other word. Therefore in this step, only a subset of the training dataset is considered:

$$\mathbf{T}_x = \{T \in \mathbf{T} \mid x = vector(T)_j, \forall j\} \tag{6.3}$$

Then, classification performances for the labeled and recorded polarity scores can be calculated and estimated. The actual performance on this subset will be:

$$\alpha(x) = \frac{\sum_{k \in \mathbf{T}_x} \#\text{ of } \{\hat{y}_k = y_k\}}{\#\text{ of } \mathbf{T}_x} \tag{6.4}$$

The substitution of $\gamma(x)$ with $\gamma(x)'$ in $\mathscr{L}$, then the predicted labels can be re-computed with this new lexicon as $\hat{y_k}' = \mathbb{F}_{\mathbf{T}}(T_k, \mathscr{L}')$. Using this $\hat{y_k}'$ instead of $\hat{y_k}$ in equation 6.4, the new performance $\alpha'(x)$ can be obtained. Next, we need to register the new polarity score before enquiring of the next word if the performance improvement surpasses a certain threshold:

$$\Delta\alpha(x) = \alpha'(x) - \alpha(x) \geq \theta. \tag{6.5}$$

If this happens, then the algorithm will try with a new ϵ. Then we implement the higher number of iterations on equation 6.2 in order to avoid the finest exploitation. In any exploitation phase, the next record T_{i+1} of the algorithm will end if the predicted label is corrected from any wrong word. Every time the sentiment lexicon is confirmed and updated, the classification algorithm $\mathbb{F}'(\mathbf{T}, \mathscr{L}')$ is retrained. Finally the next polarity records will be predicted by this new classifier $\mathbb{F}'$.

6.3.3 Convergence Constraints

The exploration-exploitation strategy is not up to the mark for providing the concurrence of diverse polarity scores. During the initial simulations, we observed the situations that no trends and alterations have been observed for some word polarities from positive to negative. In theory, two sources represented the errors in updates: (1) for polarity exploitation, different wrong words were identified, and (2) the equation is described by the situation like (6.6) that shows the continuous details for the conjugate or jointed words, i.e., a small change in one word resulting the huge effects on major performance drop for the new word.

Regardless of how, the learning process of humans for sentimental words is stable and consistent, because of the past experiences with the words stored in the memory of humanity. As time passes, the insecurities regarding the sentimental words dismiss. Due to this, we humans can clarify the polarity of words in short duration. The following convergence constraint can calculate this:

$$\zeta \in [-1/\text{count}(x),\ 1/\text{count}(x)] \tag{6.6}$$

In this equation, the $\text{count}(x) \in \mathbb{Z}$ calculates the number of polarity scores of x and has been updated till now. This calculation includes both times of exploitation and inter-instance exploration, while the convergence constraint focuses on the deepest calculation for polarity score of word x.

6.3.4 Consistency Constraints

In different domains where words alternate their sentiment, orientations are not much important for us. Past research [70] evaluated the diachronic perspective, which is related to the fact that only a few sections of words are linked with the changing in polarity for comparable long history. The alteration of the polarity of words is usually not associated with shifting of another small sentiment, especially across relevant domains, because the use of language is similar. Therefore, it is an essential requirement for checking the knowledge integration after the exploration-exploitation phase. Hence, the set required to be checked is linked with polarity switching that does not have any relation with the classification performance, that is

$$\mathbf{x}_s = \{x \in \mathscr{L}' \mid \gamma(x)'\gamma(x) < 0 \cup \alpha'(x) < \eta\} \tag{6.7}$$

where η calculates the expected performance level. Afterward, for every word $x \in \mathbf{x}_s$, the algorithm deals with another exploration-exploitation stage that depends on $\mathscr{L}'$.

6.3.5 Dealing with Negators

It is a phenomenon that keeps on appearing at different levels of natural language [36, 203]. However, it is not easy to automatically identify the scope of negation [57]. Generally, the responsibility of negators is to categorize between functional words, e.g., *no*, *not*, *never*, and *seldom*, and content words, e.g., *destroy*, *prevent*, etc. Hence, it can be said that all content words are linked with some polarity with themselves; therefore based on this fact, for vectorization records, we only deal with function-word negators. We apply the simple rule of reversing the output of $\mathbb{F}(\cdot)$ when we have detected the single function-word negator.

6.3.6 Lexicon Expansion

It has been evaluated that those target domains were characterized by words with dissimilar polarity scores and were also identified as neologisms. This happens when the environment is web-based, e.g., micro text of any new condition.[2] The allowed supervisions of expression-level or sentence-level, like in tweets, the sentiment lexicon, are mostly not present in a record like

[2]Microtexts are terminologies or short forms that are mostly not present in standard form of English but can be used for communication purposes via online sources, such as “c u 2mrw” (see you tomorrow), “abt” (about), “btw” (by the way), etc.

$$\mathbf{T}_e = \{T_i \mid x \notin \mathscr{L}\ ,\ \forall x \in T_i\} \tag{6.8}$$

As shown above, it is clear that some polarity words are absent from the records of the lexicon. The lexicon expansion algorithm initially identified and checked the POS for the label of the single word present in the records and helped in the addition of only *nouns*, *verbs*, or *adjectives* to the lexicon sentiment.

For the estimation of the polarity score of newly added letter x, a trial and error learning is used by taking in view of $\mathbf{T}_x$. Assume $\mathbf{I}(pos)$ is the positive records for the total number in the $\mathbf{T}_x$ and $\mathbf{I}(x, pos)$ denotes the frequency of word x appears in the positive records of $\mathbf{T}_x$. The calculation of polarity score of word x can be performed as a regularized difference of point-wise mutual information (PMI). This trial and error method is very popular for the automatic induction of lexicon polarity scores [117, 179].

$$\begin{aligned}\gamma(x) &= \tanh(\mathrm{PMI}(x, pos) - \mathrm{PMI}(x, neg)) \\ &= \tanh\left(\log_2 \frac{\mathbf{I}(x, pos) \cdot \mathbf{I}(neg)}{\mathbf{I}(x, neg) \cdot \mathbf{I}(pos)}\right)\end{aligned} \tag{6.9}$$

In addition to equation 6.9, we have also used different techniques of regularization and smoothing in order to divide by zero. Afterward, similar mechanisms would be used for the polarity scores of newly added words as discussed above if it needs to be activated again and again. In experiments, during the adaptation stage, this lexicon expansion is observed for all the real words of lexicons.

6.3.7 Boosting and Algorithm

As discussed at various stages that randomness is present in assignments of polarity scores, the errors recorded in the latter form of the lexicon are different in all hit and trials. As a result, the average polarity values can be determined by stochastic shifts of polarity scores of the final lexicon from diverse experimentation. At the same time, deterministic polarity shifts have been saved for a short duration for augmentation. Lastly, we represented Algorithm 6.1 termed *cognitive-inspired domain adaptation with higher-level supervision* (CDAHS) as follows.

The complexity of the implementation of Algorithm 6.1 is not based on hit and trial learning and is based on different parameters. It is clear now that the complexity is directly related to the iteration times n of boosting. On the other hand, the online calculating cost of training and implementing $\mathbb{F}_{\mathbf{T}}(T, \mathscr{L})$ can be diverse, that is, a great deal during computational analysis. For example, the SVM implementation has been taken as an example, and the solution of the inverse kernel matrix is one of the worst training cases, which is $\mathscr{O}(N^3)$. The estimation of empirical training complexity as $\mathscr{O}(N^2 D)$ and discriminate complexity as $\mathscr{O}(D)$ [21, 152], where N is the related to an initial number of records for training, and D is the lexicon size. Then, let t be the average assigning time under the consistency constraint.

Algorithm 6.1: CDAHS algorithm

Data: sentiment lexicon $\mathscr{L}$, training dataset $\mathbf{T}$
Result: adapted sentiment lexicon $\mathscr{L}'$

loop for n times ▷ Boosting
  train $\mathbb{F}_{\mathbf{T}}(T, \mathscr{L})$;
  for $T_i \in \mathbf{T}$ **do**
    if $T_i \in \mathbf{T}_e$ && $x \in T_i$ **then**
      if $POS(x) = NN \,||\, VB \,||\, JJ$ **then**
        $\mathscr{L} \leftarrow [x; \gamma(x)]$; ▷ Lexicon expansion
      end
    end
    if $\mathbb{F}_{\mathbf{T}}(T_i, \mathscr{L}) \neq y_i$ **then**
      for $x_{ij} \in T_i$ **do**
        while $\Delta\alpha(x) < \theta$ && $j < |vector(T)_i|$ **do**
          if $count(x_{ij}) < M$ **then**
            $\gamma(x_{ij}) \leftarrow \gamma(x_{ij}) + \zeta$; ▷ Exploitation
          else
            $j \leftarrow j + 1$;
          end
        end
        $\mathscr{L}' \leftarrow \gamma(x_{ij})$; ▷ Lexicon update
      end
    end
  end
  $\mathbf{x}_s \leftarrow$ comparing $\mathscr{L}$ and $\mathscr{L}'$;
  do line 9 to line 20 **for** $x \in \mathbf{x}_s \cap T_i$;
end
return $\bar{\mathscr{L}}' \leftarrow \sum(\mathscr{L}'/n)$;

Apparently t is a function of parameter θ. The time complexity for one loop is $\mathscr{O}(N^2D + D(\frac{D}{2} + 2D \cdot t) + D)$. Therefore, the overall implementation complexity is approximately $\mathscr{O}(nN^2D + nD^2t)$.

6.4 Data Description

For the targeted domain and adaptation domains, there are four original sentiment lexicons experimented with: *apparel, kitchen, electronics, healthcare, theatre, and finance* are mentioned as follows:

- Opinion Lexicon [76], it contains around 6,000 positive and negative terms for the famous list of words. It also contains words which are wrongly spelled on social media.
- SentiWordNet [5] is a form that assigns almost 117,000 values for the continuous sentiments; in English standards, it is a subset of the WordNet lexical database. For example, diverse synset may contain multiple scores. For the solution of this problem, we amalgamate the sentiment scores for all the POS label entries under the similar word when using this lexicon resource.

Table 6.2 Statistics for domain-specific datasets [194]

Domain	Apparel	Electronics	Kitchen	Healthcare	Movie	Finance
Positive	1,000	1,000	1,000	1,000	5,331	16,881
Negative	1,000	1,000	1,000	1,000	5,331	4,866
Unlabeled	7,252	21,009	17,856	5,225	0	33,579

Table 6.3 Examples of record in *finance* domain [194]

Record 1	$AAPL \xe2\x80 \x99s High Price Makes It a Risky Bet http:// ewminteractive.com/...
Clean-up	Apple high price makes it a risky bet
Record 2	$AAPL needs to chew thru trendline rez & amp; build value in this area b4 resuming higher imho
Clean-up	Apple needs to chew through trendline reservation and build value in this area before resuming higher in my humble opinion
Record 3	Couldn't take any more of Bobbie's useless drivel
Clean-up	Couldn't take any more of Bobbie's useless drivel

- L&M [103] is the most famous sentiment word list in the financial domain. According to the analysis of financial statement corpus, a polarity dictionary is composed. For this purpose, we have used all the 354 positive words and 2,349 negative words that are mostly used in financial documents.
- SenticNet [28] consists of not only word entries but also multi-word-concepts. The most recent version—SenticNet 5 has over 100,000 new data entries, each contains sentiment activation information according to the hourglass model [27] and an overall polarity score.

The supervision for the first four domains is obtained from the Multi-Domain Sentiment Dataset v2.0[3] [16]; for the *Movie* domain from sentence polarity dataset v1.0[4] [129] and for the *finance* domain from the Stocktwits dataset[5] we collected. See Table 6.2 for details.

The dataset for the *finance* domain is very demanding for its noisier nature, and number of sentiments are presented by high values and digits that require a piece of practical information for its complete understanding. Hence, apart from the removal of stop word and lemmatization, which is not necessary to perform for all the datasets,[6] we moreover remove URLs, non-ASCII characters, and hashtags and substitute some microtexts and acronyms. Table 6.3 provides examples of this cleanup prepossessing.

[3] http://cs.jhu.edu/~mdredze/datasets/sentiment

[4] http://cs.cornell.edu/people/pabo/movie-review-data/

[5] See Appendix B.

[6] Implemented with NLTK.

6.5 Experiments

The labeled data in Table 6.2 are used for experiments. The similar values for the negative and positive scores are used because of the fact that it is not possible to deal with the unbalanced data for calculating the accurate result. Specifically, a linear SVM is performed with squared-hinge loss function as an algorithm for the sentiment characterization. That is to optimize

$$\min_{\mathbf{w},\, b} \quad \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w} + \sum_{i=1}^{n} \ell[\, y_i(\mathbf{w}^{\mathsf{T}} vector(T)_i + b) - 1\,] \tag{6.10}$$

where $\mathbf{w}$, b are hyperplane parameters and

$$\ell(t) = \begin{cases} (1-t)^2, & t < 1 \\ 0, & t \geq 1 \end{cases}$$

Other characters are set as $M = 10$, $\eta = 60\%$, $\theta = 0.01$, $andn = 5$. 3-fold; cross-validation is performed to report the average training in terms of classification and stability of the accurate result. In particular, Algorithm 6.1 is benchmarked with two existing approaches: (1) TF-IDF [110], In the theory, sentiment lexicon does not present in the upper bound for all the bag-of-words because of the lack of any past information about word sentiment; (2) the automatic induction of lexicon polarity scores (AIPS), which was the "state-of-the-art" is one of the best methods dealing with Internet short texts [117]. The consequences for different domain and lexicon comparisons are present in Table 6.4.

6.5.1 Interpreting Results

In Table 6.4, the first row contains the classification accuracies of using the real sentiment lexicons, and the second row contains domain adaptation. It is a point to take in consideration that the "state-of-the-art" method is not fast and quick for various other domains. In the *finance* domain, it works more appropriately, because of the fact that the dataset consists of short texts and is under huge supervision. Regardless of this, its performance is not robust for other domains. For example, in the *health* domain, the random guess is more accurate than using the calculated polarity scores. In opposite, TF-IDF can draw reliable results, even though this method does not produce any sentiment lexicon.

It is worth mentioning that whatever sentiment lexicon is taken as a starting point for Algorithm 6.1, it can always outperform the TF-IDF baseline. No negative score is estimated, i.e., after the domain adaptation, the results always move toward accuracy. This is the main difference between Algorithm 6.1 and other transfer

Table 6.4 Sentiment classification accuracies for six domains, showing competition before/after domain adaptation. (Adapted from [194])

	Apparel	Electronics	Kitchen	Healthcare	Movie	Finance
TF-IDF	74.2%	66.0%	65.0%	65.0%	75.4%	68.1%
AIPS	54.5%	53.3%	51.3%	49.0%	53.1%	**71.7%**
Opinion Lexicon	66.2%	64.2%	62.5%	60.3%	69.4%	58.0%
	72.8%	**69.2%**	**69.7%**	**66.5%**	74.7%	65.6%
SentiWordNet	66.5%	63.2%	59.3%	59.2%	68.4%	57.7%
	71.0%	65.9%	64.1%	64.2%	75.2%	63.6%
L&M	63.2%	60.0%	61.5%	53.2%	58.0%	54.0%
	70.5%	64.2%	68.3%	62.7%	69.1%	62.0%
SenticNet	70.5%	64.8%	60.5%	63.2%	71.0%	62.7%
	74.7%	**69.2%**	69.3%	65.7%	**77.9%**	69.8%

learning based procedures. SenticNet is one of the best methods without domain adaptation and provides average classification accuracy that is 65.5%, come after the Opinion Lexicon (63.4%) and SentiWordNet (62.4%). L&M is a domain-specific lexicon; therefore, it is very clear that its functionality is the lowest (58.0%) for diverse domains.

After the domain adaptation process, the gaps flanked by dissimilar lexicons are tapering. SenticNet is still somewhat improved (71.2%), pursued by of. Opinion Lexicon (69.7%), SentiWordNet (67.4%), and L&M (65.8%). Algorithm 6.1 perk up L&M (7.8%) and Opinion Lexicon (6.3%) additional to SenticNet (5.7%) and SentiWordNet (5.0%). This is most likely since the previous two have fairly limited vocabulary extent. As a consequence, more vocabulary is supplementary in the lexicon spreading out stage with dataset-induced polarity attained. This may also entail that a best possible number of kernel words subsist: a steadiness amid former information and the facility to adjust in a new domain.

6.5.2 A Showcase for Sentiment Shifts

Mounting emotion lexicons with appliance knowledge-based methods typically bring in the difficulty of over-fitting. Vocabularies that hold no feeling are allocating division scores due to the accidental statistical inequity. These lexicons might carry out very fine on the preparation dataset, but do not craft sense and have deprived simplification aptitude. To look at whether the version of word polarities in Algorithm 6.1 is reliable with practical information, the words that most intensively distorted their polarity scores are recorded. Figure 6.4 offers the necessary information. To make the analysis brief, only the adaptation results using SenticNet as the starting sentiment lexicon are indicated here because of its good performances both before and after domain adaptation.

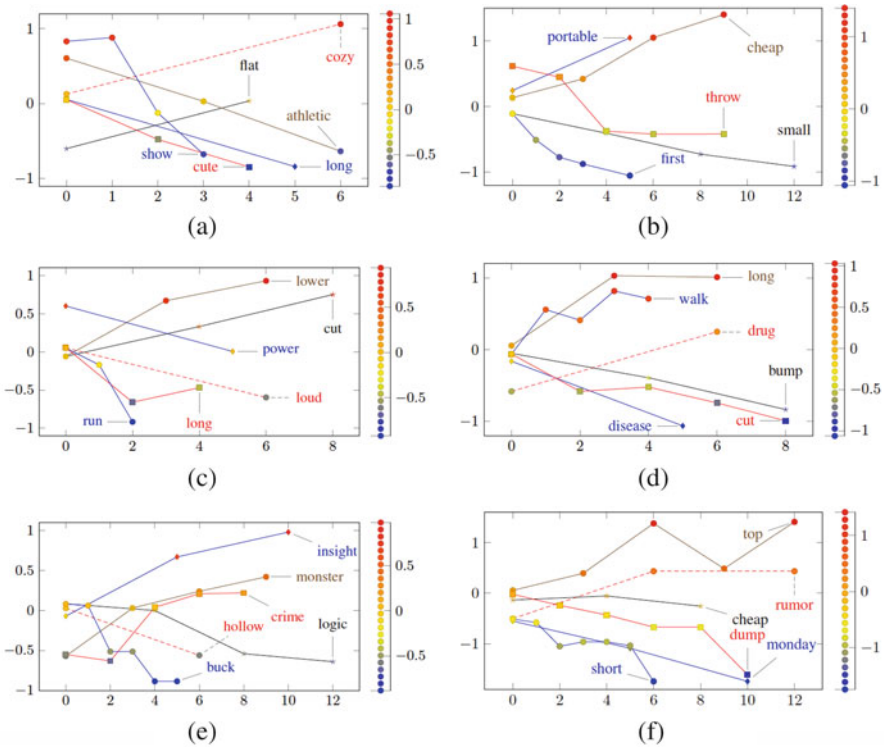

Fig. 6.4 Sentiment shifts of words in different domains [194]. (**a**) Apparel. (**b**) Electronics. (**c**) Kitchen. (**d**) Healthcare. (**e**) Movie. (**f**) Finance

Numerous instances offered here are confirmed by the preceding studies: *war*, *dark*, and *complex* are positive descriptions for movies [111]; *easy* is typically used for positive assessment in the *electronics* domain, e.g., *easy to use*; on the other hand, it is negative in the *movie* domain [185]; *unpredictable* is affirmative in the *movie* domain, e.g., *the plot of this movie is amusing and changeable*; still, it is a negative expression in the *kitchen* domain [186]. All these statements are justified in Fig. 6.4 and more similar instances exist throughout the domain adaptation process.

The trials propose that a lot of speech polarities move from approximately unbiased to conflicting guidelines in different domains. For instance, *cheap* is unbiased in the universal domain. However, it changed to positive in the *electronics* domain, because it is a desirable property for clientele. In the *finance* domain, in distinction, people do not like *cheap stocks*, so the polarity turns out to be somewhat negative. Likewise, *long* is positive in the *health* and *finance* domain, e.g., *long life* and *long position*. However, it is off-putting in the *apparel* and *kitchen* domain. Words *crime* and *monster* are more often than not looked upon as negative in the current domain. However, in the *movie* domain, they pass on to sort as an alternative of their main meanings; *power* does not pass on to power or ability,

but the foundation of energy in *kitchen* domain. Various less instinctive examples are linked with jargons and words usage. For example, *Monday* is impartial in the universal domain. Nevertheless, marketplace crashes and liquidity harms are more likely to take place on Monday [3, 190], e.g., *Black Monday*. As a result, its division happens to be very negative.
Word *logic* changed to negative in the *Movie* domain, not because of the feeling of its nuance. Instead, the reason is that to say "*I dislike the movie because it doesn't have any logic*" is natural and extra probable than to say "*I like the movie since it has logic*." Likewise, when citizens talk about the rumor in the *finance* domain, they usually entail a piece of information that could make an income, not awful reports that could cause defeat.

Another motivating tip to observe is that in the *movie* and *finance* field, statement polarity achieves and alters more era before the junction. Since these two fields have quintuple dimension of other pasture in terms of the number of records, it is reasonable to believe a more substantial training data enables a more accurate search for word polarities. When supervision is weak, the adjustment takes more significant steps and does not suffice to correct the small deviation from the causal word division.

6.6 Summary

This chapter has shown a blueprint about how semantic and sentiment knowledge can be stored and updated. This does not belong to the central asset allocation part but is of equal importance for a knowledge-based financial asset management approach to work and to be maintained. As an initiative, a cognitive-inspired algorithm is proposed to adapt sentiment lexicons to the target domain. The sequential learning algorithm is almost passive, regardless of how the results for the performances are compatible with some questionable strategies for the collection of information to learn the information. A promising extension of the algorithm is to learn polarity for concepts, instead of words.

In particular, these sentiment adaptation techniques have appropriate characteristics, i.e., it is robust, no negative learning occurs, and it presents an updated sentiment lexicon for the specific domain, which embraces high interpretability. This sentiment lexicon naturally serves as a fundamental form of the sentiment knowledge base.

Chapter 7
Robo-Advisory

I'm telling you. Who's on first, What's on second, I Don't Know Who is on third.

— Abbott and Costello

Abstract Robo-advisory completes the last missing part of the vision of intelligent asset management—featuring the human-computer interaction process that provides the necessary information for the asset allocation algorithms. In this chapter, we picture the industry by studying companies that do robo-advisory as a service. The main body discusses the technical framework of a robo-advisor, how it is different from the traditional financial advisory process, and the latest relevant research about dialog systems and recommendation systems. In the end, we develop the outlook for the future of robo-advisory.

Keywords Financial advisory · Dialog system · Recommendation system · Decision support · Digital assistant

Chapters 4, 5, and 6 describe the main techniques underlying AI-empowered asset allocation strategies and the infrastructure knowledge base. Though in real-world cases, the whole process of asset management cannot be accomplished without the proactive participation and involvement of the investors. In delegated asset management, the investors are clients who deposit their capital to the management team. Robo-advisory mainly deals with the communication process between clients and the investing agent. According to the CFA Institute (Chartered Financial Analyst), robo-advisors are online platforms that provide automated investment advice based on a customer's answers to survey questions [150]. Jung et al. define robo-advisors as digital platforms comprising interactive and intelligent user assistance components that use information technology to guide customers through an automated investment advisory process [81]. A more detailed definition by Day [44] emphasizes that robo-advisory often uses algorithms with the characteristics of low cost, availability, and ease-of-use. However, the core idea of customizing the investment according to clients' requirements remains unchanged. Sironi [156]

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4_7

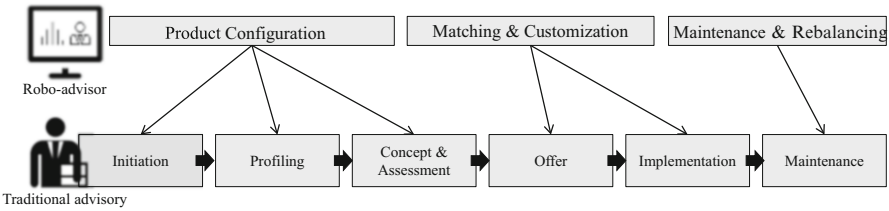

Fig. 7.1 Mapping between robo-advisory and the traditional financial advisory process. (Adapted from [81])

describes the process straightforwardly as "translating the client's" specific needs into an adequate portfolio of financial products." In the Markowitz model, this information is compressed as a single parameter: the risk aversion coefficient. The only evolution is that the process is digitalized and conducted in a more smooth and convenient way.

Jung et al. [81] conceptualize the degree of digitalization with two waves: the first wave brings the brokerage platform online, and the second wave makes the process intelligent. Moulliet et al. [47] elaborate it with four stages. In the first stage, the communication is facilitated by online questionnaire and proposal of candidate portfolios for the client to choose. In the second stage, the algorithms will help the client with automatic adjustment of positions and rebalancing. In the third stage, the algorithm will not only manage the portfolio but also explicitly convey the rationales to the client, for example, pre-defined rule sets. In the final stage, the algorithm will have more advanced features such as self-learning and asset shifts not only inside each class but across different asset classes. Figure 7.1 shows the mapping between robo-advisory and the traditional financial advisory process, illustrating the gradually simplified stages of services. In the traditional settings, the advisor first prepares a meeting with her customer (initialization). During the meeting, the advisor talks to the customer to get necessary information (profiling) and develops a concept. The three phases are combined in robo-advisory. Based on the concept, a human advisor would make an offer out of the investment universe: if the customer accepts the offer, it will be implemented with capital from the customer. In robo-advisory, this phase can be automated by matching a popular portfolio for customers of-this-kind or customizing a new composition. In both cases, the advisor will have to keep in contact with her customer and explain or ask for approval for vital changes to the investment classes.

In fact, the simplified service is an adaptation to a key feature of low-cost robo-advisory compared to the traditional financial advisory service. This is also an important motivation for developing such systems. As a practice to financial inclusion, robo-advisory makes an alternative to the expensive wealth management service with higher efficiency and accessibility to common people with small savings. Although these customers have not shown strong interest in robo-advisory, it tends to become the only possible and economic way for them to participate in financial investment [82]. This research also gives concrete guidelines for designing

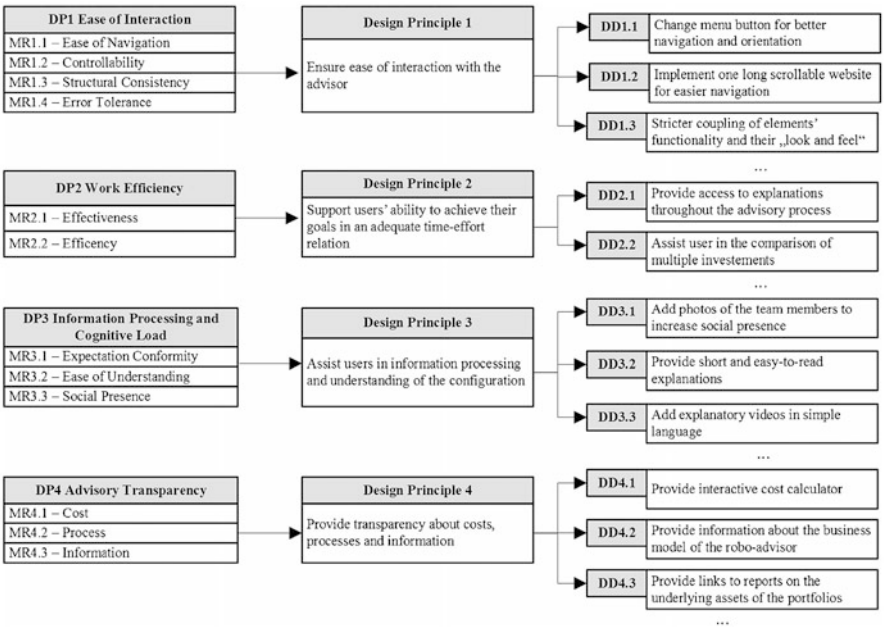

Fig. 7.2 Design principles for a robo-advisor [81]

the user interface for asset allocation algorithms. Figure 7.2 breaks down the four design principles. In the next section, we will see how well the robo-advisory companies follow the principles and design their products. Analysis from aspects like the fee structures and minimum amount of investment shows that robo-advisory generally cuts 70% to 80% of the cost compared to the traditional methods.

7.1 Industry Landscape

We study some famous robo-advisory companies in the USA, China, Europe, and Singapore (see Table 7.1). The US robo-advisory industry thrives after the 2008 Financial Crisis due to risk aversion and that having a portfolio was a relatively robust investment tool. Two flagship companies Betterment and Wealthfront, for example, were established targeting at middle-class customers with 200 to 300 thousands annual income. Nowadays, the business has a market share of over 300 billion and is still fast growing according to Statista Market Forecast. Robo-advisory in China started off from 2015, but already reached a scale of 90 billion USD asset managed. Other representative companies are from regions with a strong financial sector.

We observe that there are two types of fees pertinent to the robo-advisory services. Annual management fee can either be a flat rate or tiered priced according

Table 7.1 Representative robo-advisory companies and their products (Data collected on 2019-04-09)

Company	Product	Country	Fee structure	Minimum amount	Assets invested
Betterment	Smart solutions	USA	0.25%–0.4% annual fee	0	ETFs
Blackrock	FutureAdvisor	USA	0.5% annual fee	5000 USD	ETFs and Stocks
Charles Schwab	Schwab Intelligent Portfolios	USA	Transaction fee	5000 USD	ETFs
Goldman Sachs	HonestDollar	USA	0.25% annual fee	0	ETFs
Personal Capital	Smart Weighting	USA	0.49%–0.89% annual fee	0	ETFs and Stocks
Wealthfront	PassivePlus	USA	0.25% annual fee	500 USD	Stocks, Bonds, Real estate, Nature resources
SIGFIG	MANAGED ACCOUNT	USA	0.25% annual fee	2000 USD	ETFs
China Merchats Bank	Mojie	China	1% transaction fee	N.A.	Funds and QDII
JRJ.com	Lingxi Zhitou	China	Transaction fee	500 CNY	Stocks, Bonds, Gold
Licaimofang	Personalized advisor	China	Transaction fee	N.A.	Funds
Swissquote	Robo-Advisory	Switzerland	0.95%–1.25% annual fee	10000 CHF	Securities
Crossbridge Capital	CONNECT	Switzerland	0.2%–1.25% annual fee	500000 SGD	Cash, Fixed income, Equities, Alternatives
StashAway	ERAA	Singapore	0.2%–0.8% annual fee	N.A.	ETFs
OCBC	RoboInvest	Singapore	0.88% annual fee	3500 SGD	ETFs and Stocks

to the amount invested or coverage of system functions. Some of the robo-advisory services do not charge management fees; instead, they charge the transaction fees generated when allocating assets. In such cases, the advisory provider has a motivation to frequently re-weight portfolios as this promotes the sale-side profit. Chinese robo-advisory services are dominated with the transaction fee mode and the rest of the world annual management fee mode. In terms of assets invested, ETFs (exchange-traded funds) are popular because they are ideal tools for the philosophy of long-term, passive, and diversified investment. ETFs are often highly liquid

and low in transaction fees and can simplify the investing process, while Chinese robo-advisory services mostly invest in funds, so the product becomes a fund of funds (FOF) or fund recommendation system. No surprise to the phenomenon given the fact that Chinese ETF market is underdeveloped on many assets such as commodities, FOREX, real estate, etc. Though the product risk becomes more unpredictable in such cases due to the subjective operations and shifts by fund managers, except ETFs, individual stocks and alternatives such as real estate, natural resources, and gold are also viable asset classes (see Table 7.1). Some companies argue that ETFs have two kinds of inefficiency. First, ETF fees are passed to the clients in a process known as "fee stacking," so it is only necessary when outright investment on certain stocks is not accessible. Second, the rebalancing cost is higher for ETFs (because of position taken) compared to individual stocks when markets move, which will impact investment returns. Furthermore, the ETFs may already adopt a suboptimal diversification, reducing the opportunity to harvest the same-class asset volatility. Another difference is that many robo-advisory services in the USA are specialized in tax-efficient portfolios and retirement plans, probably because of the complicated US taxation [77], whereas in other regions, companies tend to mention more about their technical advantages. For example, Lingxi Zhitou discloses techniques underlying their investing algorithms, including constrained Black-Litterman model, the utility theory, and an asset weighting chart. Economic Regime-based Asset Allocation (ERAA), a registered method brand of StashAway on RBAA [125], detects the current economic regime and re-optimizes its investment portfolio return with a constant risk when the economic environment changes. The idea is similar to the macro-factor investing model and Bridgewater's "All Weather" strategy [2]. These characters are more popular on volatile markets.

There are huge differences for the companies' capabilities of profiling their clients. Traditional financial advisory uses questionnaires to assess the client's mindset and experience in investing. An instance is the Investment Personality Assessment by Merrill Lynch,[1] which discusses financial resources for family, lifestyle, goals, priorities, etc. For robo-advisors, a rather simple product is Mojie by the China Merchats Bank: the robo-advisor only takes two inputs from the client, i.e., the investing time horizon (years) and the risk preference (scale from 1 to 10). However, most other products look at more dimensions. For example, Lingxi Zhitou will require client information such as age, family members, financial position, risk tolerance level, and the target return so as to estimate the cash flow in the upcoming years. The information is gathered through a carefully designed questionnaire to validate its consistency and will be used to assess subjective risk preference and objective risk tolerance. Overconfidence is a well-established cognitive bias in investing. Therefore, the final risk level will be set on the conservative side of the preference/tolerance spectrum.

The analyzed industry products mostly achieve the first or second stage of digitalization [47]. To push forward the frontiers, researchers identify two fields in

[1] http://www.ml.com/life-goals/investment-personality-assessment.html

artificial intelligence that work on refining the configuration and matching phases of a robo-advisor as in Fig. 7.1. Dialog system can be used to acquire user information, and recommendation system leverages this information to select from a pool of candidate portfolios.

7.2 Robo-Advisory and Dialog System

Although dialog system has become a hotspot research direction in NLP in the recent 3 to 5 years, few of them are applied to build a robo-advisor. There are several reasons for it. First, dialog systems are mainly categorized to task-oriented dialog systems and chatbots. The technical architecture and objective of the two types are quite different. A task-oriented system has state trackers and policies for response selection, while a chatbot uses neural generative models learned from datasets [34]. Robo-advisory, by its nature, is task-oriented. However, the experience with task-oriented systems is inferior in terms of topic- and discourse-level coherence and interactiveness compared to generative models. Therefore, there are not enough incentives to migrate from the traditional questionnaire to a task-oriented system. Second, the industry did not realize to collect such conversational data due to privacy and cost concerns. The past interactions between financial advisors and clients are also concentrated on high-value customers. The target middle-class group was not involved in this service; thus the lack of data can be expected.

We believe the backbone model for a robo-advisor should be generative to allow some flexibility. One option is to have a parallel task-oriented system and switch between the two. Another option is to somehow integrate external domain knowledge. As far as we know, Day et al. [45] was the first to attempt to combine an asset allocation model with a standard Seq2Seq model to build a prototype robo-advisor. Figure 7.3 shows the system architecture.

We see that the system uses a large corpus (STC_weibo) as the base to training a Seq2Seq model and evaluates the model with a self-constructed financial question answering dataset. Though there lacks a way to actually improves the training model, it is integrated with asset allocation strategies to social media platforms. Hopefully the conversational model still captures some information so that the asset allocation strategy can be accordingly adjusted. The sequence-to-sequence model (Seq2Seq) is the most widely applied neural architecture for dialog generation. Proposed by Sutskever et al. [163] firstly for the machine translation task, the model maps the dialog context to a sequence of words in the generated response. Given a dialog context $X = x_1, x_2, \ldots, x_N$ consisting of N words, the model outputs a responsive sequence $W = w_1, w_2, \ldots, w_M$ of length M via a hidden context representation $H = \alpha_1 h_1 + \alpha_2 h_2, \ldots, \alpha_N h_N$, where each $h_t = \phi(x_t, h_{t-1})$ and α_i are attention weights normalized over the input sequence so that they sum up to 1. A Seq2Seq model has a RNN to map the function $X \rightarrow H$ called encoder and another RNN to map the function $H \rightarrow W$ called decoder. Hence the objective of the Seq2Seq model can be defined as:

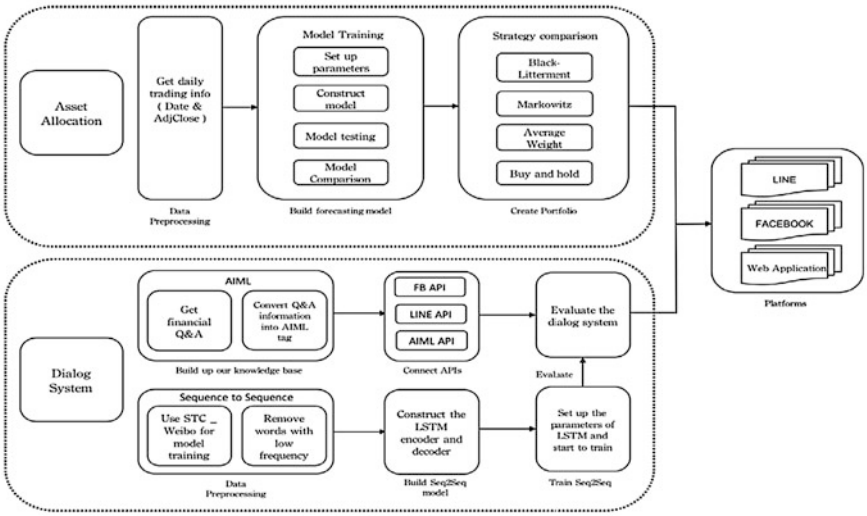

Fig. 7.3 System architecture of a conversational robo-advisor proposed in [45]

$$\text{argmax} \prod p(w_t|X, W_{-t}) \tag{7.1}$$

where w_t is drawn from a distribution over vocabulary based on H.

Another useful direction for robo-advisory is personalized dialog systems. The related business scenarios are digital companions, such as in-car assistant, restaurant booking, and well-being chatbots. In these scenarios, the dialog agent has to store some features of the human-being that she has a long-term conversation with. To enable this, user personality modeling via dialog history is helpful. The personality models that have been used are Big Five personality traits[2] and the MBTI theory. For example, Fung et al. [63] developed an interactive dialog system that fuses multimodal information (visual, speech, and textual) to detect the user's personality based on the MBTI theory.[3] The theory categorizes human personalities into 24 types using 4 dimensions, i.e., introversion vs extroversion, intuitive vs sensing, thinking vs feeling, and judging vs perceiving. Figure 7.4 shows a screenshot of the system panel.

The central problem in interaction with a robo-advisor lays on risk aversion estimation, which has a tight connection to personality. Expressing the same level of risk-aversion, an introversive person may be more vulnerablewhen high volatility

[2]Also known as the OCEAN model [142], which exams five factors: openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism.

[3]http://www.myersbriggs.org/my-mbti-personality-type/

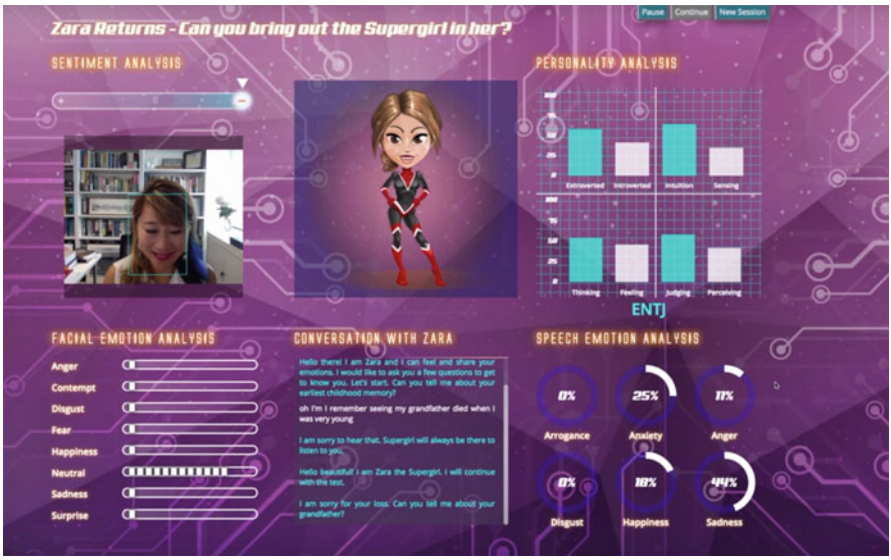

Fig. 7.4 The system panel of "Zara", a dialog system that detects human personality

events occur. Thus the asset management strategy customized should be more conservative. How to quantify this adjustment is an interesting topic and worth researching.

7.3 Robo-Advisory and Recommendation System

The ideal asset allocation module manages an asset pool specifically customized for each client. However, because managing a large number of assets is a challenging optimization task, companies usually like to downscale the portfolio in two ways. The first way is to invest in ETFs instead of individual stocks. The second is to formulate some "ready-made products" in advance and pick one of the most suitable for each customer. A tech-savvy portfolio, for example, may invest a high weight on the US tech companies such as FANG Stocks[4] and is characterized by high expected return and high volatility. In both cases the module reduces its cost by selecting from a finite number of passive asset options, therefore many robo-advisory products are actually recommendation systems, see Fig. 7.5 for an example.

The early purpose of using the recommendation techniques was to sale to customers different types of financial services, for example, loans with certain amounts [59]. Although not in the context of recommending assets, the VITA

[4] https://www.investopedia.com/terms/f/fang-stocks-fb-amzn.asp

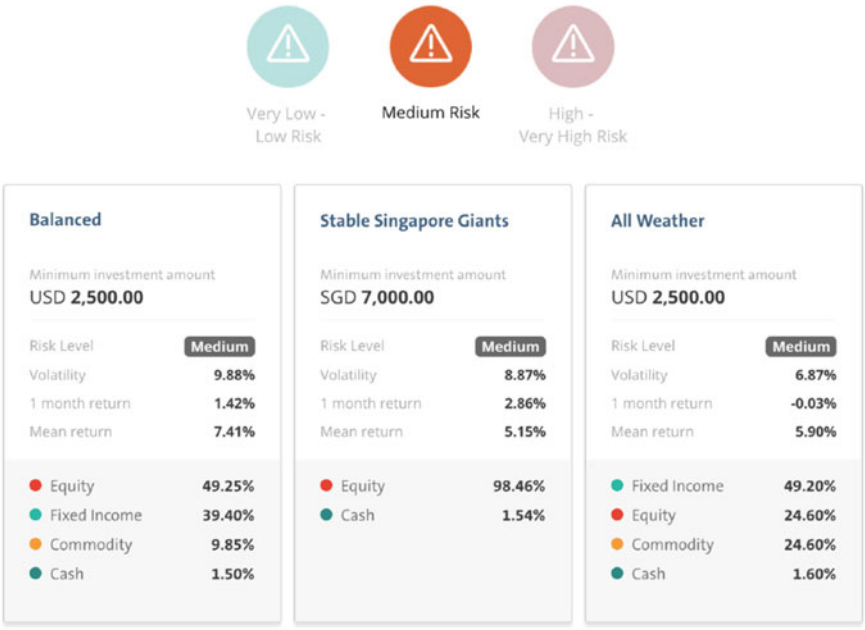

Fig. 7.5 The candidate portfolios to choose from at RoboInvest of OCBC

application uses a content-based recommendation system. For many dimensions of a product, the information provided by each customer will be used to rate the product according to the multi-attribute utility theory [147]. Therefore, a subset of products that meet the customer's requirement will be ordered and presented. Nowadays, recommendation systems use two approaches or a hybrid of them. The collaborative filtering approach leverages the knowledge of user representation. Users are defined by their past behavior or their related characters. The system will recommend items liked or purchased by the user's peers; thus the problem is converted to measuring user similarities or discovering similar users (through clustering). The content-based approach leverages the analysis of the items themselves compared in a unified framework. As a result, all the recommended items will be alike.

Robo-advisory has access to both item attributes and user-centered information, such as family and income structures. Therefore, the recommendations are made with a consideration of the customer's social relations. Financial social networks are complicated, and the nodes can be divided into several groups. The data used by Xue et al. [196] contains three groups: investors, financial institutions, and the market environment. Then, to measure the similarity between two investors, the system will consider user ratings together with the graphical structure, which is implemented by solving the rating matrix factorization problem with social regularizations [105]. The research also experimented with three strategies of model aggregation for group recommendation, i.e., default, average, and least misery strategy. The results suggest

the differences are not significant. With the group recommendation algorithm and the introduction of a financial social network, the overall expected return and maximum drawdown both decrease, perhaps showing that the recommendation in general improves the portfolio for customers of all ranges of risk preferences.

7.4 Robo-Advisory and Active Investment

Being a cost-efficient investment channel, many robo-advisory products in Table 7.1 only charge 0.25% to 0.8% flat fee. This makes active management of assets challenging. However, there are still more than half of the surveyed products in Germany that provide active management options [47]. If the product is shaped to provide high returns, the asset allocation model will have to be aware of the market environment changes and jump from one weighting to another. Subsequently, the major portfolio components do not have to be highly risky. In this situation, asset risk premiums are substituted with the decision-making competency of the asset allocation model, and the robo-advisory service itself becomes a derivative building on the assets in its portfolio. Meanwhile, the ideal targets for active investment are well-diversified, fixed-risk, and easy-to-trade assets due to cost constraints, such as ETFs.

Passive investment, on the contrary, allows to include a wider range of assets. PassivePlus by Wealthfront in Table 7.1 is a self-explanatory example that invests even on some very illiquid assets such as real estate and natural resources, which requires the asset allocation model to accurately estimate asset expected returns and volatilities for a longer horizon. An important feature of asset allocation models is the frequency for rebalancing. Most of the current industry products cope with low-frequency models for multiple reasons. First, building trust between the robo-advisor and its customer is even more difficult than between a human financial advisor and her customer. Highly frequent strategies need to brief their customer very often, which incurs additional cost. Second, in order to differentiate the product from hedge funds and target for middle-class customers, the robo-advisor aims at long-term growth in analogy with the market growth. So there is a strong motivation to reduce transaction fees by introducing inertia. We observe several products that rebalance four to six times per year, which allows quarterly reassessment of the customer's risk preference change.

Chapter 8
Concluding Remarks

We are forced to act largely in the dark.
—Fischer Black

Abstract In this final chapter, we summarize the whole book by reviewing concepts and algorithms proposed, as well as theories derived in this book. An outline of extracting and leveraging different aspects of knowledge from financial texts with the help of NLP techniques is given. We also mention here the model limitations, the issue of data availability, and statistical power of simulation results. Promising future directions of this research topic are also discussed at the end of this chapter.

Keywords Data availability · Market liquidity · Knowledge integration · Financial text · Deep learning · System deployment

8.1 Concepts, Algorithms, and Theories Derived

Important novel concepts proposed in this book are:

1. **Natural language-based financial forecasting**: Forecasting activities on financial indicators, such as asset return, risk, and holding position using NLP techniques and financial texts, mainly computational semantics and sentiment analysis of financial reports, company releases, and social media data streams.
2. **Narrative space for financial information**: A dichotomy of semantics and sentiment. A perspective that separates invariant correlations between financial assets from the temporal volatile components.
3. **Semantic vine**: A regular partial correlation vine (dependence structure), where the partial correlation value of edges is estimated from pairwise semantic linkages between financial assets as nodes.
4. **Market sentiment views**: Market views of the Black-Litterman model, where the investor's subjective expected returns of assets are approximated by time series of prices, trading volumes, and sentiment from social media.

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4_8

5. **ECM-LSTM**: A novel RNN design inspired by an online clustering method and deep learning that filters learning instances to increase stability and avoid the overfitting on the time axis.

Algorithms developed are:

1. **Growing semantic vine structure** (Algorithm 4.1): The process of sequentially adding edge-spanning pairs to a group of financial assets providing their pairwise semantic linkage.
2. **Estimating robust correlation matrix** (Algorithm 4.2): The process of estimating a robust correlation matrix (risk indicator) for a group of financial assets based on the semantic vine structure.
3. **ECM-LSTM training and forecasting** (Algorithm 5.1): An online procedure that deeply combines ECM and LSTM training and output and predicts the subjective expected returns of assets.
4. **Cognitive-inspired domain adaptation with higher-level supervision** (Algorithm 6.1): A supervised algorithm that searches polarity scores for words in a lexicon and expands the vocabulary by adding new sentiment words.
5. **Augmented sentic patterns**: An extension that enables sentic computing to produce not only a positive or negative label [136] but also an intensity value of sentiment between -1 and 1.

Theories derived are:

1. **Theory of development stages of NLFF** (Fig. 1.1): A theory that asset allocation models, as the second wave, would supersede the price prediction paradigm when an investor considers heterogeneous assets and finally lead to intelligence in financial asset management.
2. **Types of financial texts**: A theory that financial texts can be categorized into six main groups—corporate disclosures, financial reports, professional periodicals, aggregated news, message boards, and social media posts—according to three criteria, text length, subjectivity, and the frequency of updates.
3. **Hierarchical structures mapping** (Fig. 3.5): A theory that there is a mapping between different ways of describing the structure of language.
4. **Human learning of sentiment**: A group of metacognition processes that emulate a human agent when the sentiment of neologisms is uncertain or unknown, e.g., exploration-exploitation, narrow down of search space, etc.

Besides, we proved some properties of market views and the formula for calculating unconditional correlation.

8.2 Limitations and Future Work

8.2.1 Limitations

As described in Chaps. 4 and 5, semantics and sentiment are separately used to model asset correlations and expected returns. A limitation of this approach is that a theoretical difficulty emerges: since the robust correlation matrix is static and estimated from a different source, it cannot be interpreted as the covariance of asset returns. This difficulty requires innovative elucidation of the mean-variance optimization framework. It is also an unanswered question whether there is a third way to integrate textual knowledge to asset allocation models—extending another dimension for the narrative space for financial information—and how to do it. Otherwise, new advances, e.g., in time expression analysis will have to be added to one of the current dimensions.

It may be easy to notice that some developed algorithms contain many hyper-parameters. These parameters have psychological connotations, e.g., risk aversion, consequently, are almost impossible to optimize with the usual methods such as grid search. We generally follow previous literature to decide these parameters. A complete investigation on the sensitivity of them would be very interesting and challenging. Another limitation is that in both the augmented Markowitz's model and Black-Litterman's model, transaction cost is not considered. The hypothesis is that the cost is negligible when the capital managed is extremely large. However, a paradox is that in this case, the liquidity problem is the other side of the coin. The investor may not be able to find a market maker to do large transactions, which is not included in any model discussed in this book. It is still unclear how and to what extent the portfolio strategies are affected by encouraging frequent rebalancing with zero transaction cost. Since most of the strategies discussed in this book adopt daily rebalancing, it may be potentially beneficial to abandon small changes in holding positions.

Data availability is always a challenge in financial applications. Since we only collected limited data, the method to inversely assess the quality of data employed is not thoroughly developed, though many researchers are interested in it. Finally, this limitation of insufficient data causes concept sparsity in the corpus, which may bring difficulties when we use the CDAHS algorithm to build a concept-level sentiment knowledge base in the future.

8.2.2 Future Work

We plan to pursue the following future work, a part of which is ongoing:

1. **Constructing a concept-level sentiment knowledge base**: An extension of the CDAHS algorithm to include not only word entries but also concepts and

corresponding polarity scores. Specifically, we have developed algorithms that automatically extract and select concepts from formal financial reports, not only in English but also in Chinese.
2. **Studying portfolio optimization with transaction costs**: This problem, in general, can be formed as an optimization problem with constraints [13]. We plan to substitute the Markowitz model with it and analyze the strategy sensitivity to cost functions. This direction has practical meaning for building robo-advisors.
3. **System deployment**: The approaches proposed in the current stage have a great potential for industry partners. We will look for applications in other research projects and collaboration with banks, funds, and asset management entities.

8.3 Conclusions

In this book, we introduced several NLP techniques to tackle problems in asset allocation models. This perspective generalizes price prediction problems when multiple heterogeneous assets are available on the market. According to the type of financial texts, we proposed to extract different aspects of information, such as semantics and sentiment. Advantages and room for improvement of these techniques are discussed and supported by experiments.

Econometric models can still be used in the asset allocation models just as their wide applications in time series analysis. However, they may suffer from issues such as instability and limitation of information sources. A major difference between applying econometric models and our machine learning-based NLP methods is that the latter usually have at least thousands of parameters doing nonlinear matrix operations. This expressive power has merit in approximating complicated market behaviors and phenomena. Moreover, despite the fact that econometric models can include exogenous variables, there exists a gap between non-structured textual data and numerical variables. NLP is still indispensable to facilitate this conversion in real time and in a large volume.

We explored both sub-symbolic AI for semantic modeling of asset correlations (Chap. 4) and a symbolic AI approach for sentiment analysis of the mass opinion on asset returns (Chap. 5). Based on a portfolio of five stocks, we find that the robust estimation of asset correlations by semantic linkages is superior to estimation using historical price data. With the help of a proper semantic vine, the portfolio outperformed 80% to 90% peers of its kind in terms of annualized return and is even superior to the market portfolio in terms of Sharpe ratio. The method only has $\mathcal{O}(\log n)$ more complexity than naïve computation and can scale up to more than 50 stocks in less than 5 seconds.

With the help of augmented sentic patterns and sentic computing, we developed a method to obtain sentiment time series on specific assets, which is further validated by strong correlations to some commercial tools and products. The same portfolio as the previous is strengthened by a novel neural network design (ECM-LSTM). The impact of adopting the sentiment time series and ECM-LSTM are evaluated,

respectively. The improvement in annualized return is circa 2% for sentiment and more than 10% for ECM-LSTM. The first percentage of contribution is backtested for 8 years; however, due to data availability, the latter one is only backtested for 3 months.

The application of AI in financial modeling is not well-realized until recent years. One of the main reasons is that there are many pitfalls that would invalidate normal practices in general machine learning problems. For example, preventing information leak on time axis needs special efforts; model training cannot shuffle data because the temporal structure is important; replicability in historical data cannot guarantee replicability in the future. Another important requirement for financial applications is the interpretability. Because the models need to be finally customized for and briefed to investors, an end-to-end system with no meaningful intermediate results is not preferred. If black box algorithms are necessary, they have to be restricted to computing known variables in the minimum-possible modules. The similar aim is also recognized by AI in healthcare and medical research.

Of course, this book does not cover all the aspects of intelligent asset management. Those materials on properties of asset types such as equities, bonds, alternatives, and funds can be found in classic textbooks of finance, while hands-on guide to incubate a viable product and regulation issues is more carefully analyzed in white papers. Even in the scope of incorporating newly developed AI techniques, there are emerging topics that have just come under the spotlight, such as the fake news problem [115]. Even on professional platforms, malicious and misleading information are accumulating, stimulating quests for algorithms that evaluate the credibility and impact of financial news. Data privacy and security is another unaddressed topic in this book. With more people from the legislative and jurisdictional background involved, more progresses in this direction are expected.

Financial asset management is not only science but also an art. Recent advances in AI and machine learning achieved a great deal in computer vision but less in NLP because language is a more abstractive mental process. Every human baby can recognize objects but not everyone has equal excellence in language ability. Financial asset management is a challenging task even for professionals. There is still a long way to go before we really understand how to achieve this type of intelligence.

Appendix A
Stock List and Vine

The full stock list used for scalability analysis is:

Number	Ticker	Company name	GICS	TRBC
0	AAPL	Apple Inc	4520	5710
1	ABT	Abbott Laboratories	3510	5620
2	AXP	American Express Co	4020	5510
3	BA	Boeing Co	2010	5210
4	BAC	Bank of America Corp	4010	5510
5	BMY	Bristol-Myers Squibb Co	3520	5620
6	C	Citigroup Inc	4010	5510
7	CAT	Caterpillar Inc	2010	5210
8	CMCSA	Compcast Corp	2540	5330
9	CBG	CBRE Group Inc	6010	5540
10	COP	ConocoPhillips	1010	5010
11	CTL	CenturyLink Inc	5010	5810
12	CVX	Chevron Corporation	1010	5010
13	D	Dominion Energy Inc	5510	5910
14	DD	DuPont (EI) de Nemours	1510	5230
15	DIS	DISNEY (WALT) CO	2540	5330
16	DPZ	Domino's Pizza Inc	2530	5330
17	DUK	Duke Energy Corp	5510	5910
18	ECL	Ecolab Inc	1510	5110
19	EXC	Exelon Corp	5510	5910
20	FTR	Frontier Communications Corp	5010	5810
21	GE	General Electric Co	2010	5230
22	HD	Home Depot Inc	2550	5340
23	IBM	Intl Business Machines Corp	4510	5720

(continued)

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4

Number	Ticker	Company name	GICS	TRBC
24	INTC	Intel Corp	4530	5710
25	JNJ	Johnson&Johnson	3520	5620
26	JPM	JPMorgan Chase&Co	4010	5510
27	KO	Coca-Cola Co	3020	5410
28	MCD	McDonald's Corp	2530	5330
29	MMM	3M Co	2010	5230
30	MO	Altria Group Inc	3020	5410
31	MOS	Mosaic Co	1510	5110
32	MRK	Merck&Co	3520	5620
33	MSFT	Microsoft	4510	5720
34	NEE	NextEra Energy Inc	5510	5910
35	NEM	NewmonT Mining Corp	1510	5120
36	NKE	Nike Inc	2520	5320
37	ORCL	Oracle Corp	4510	5720
38	OXY	Occidental Petroleum Co	1010	5010
39	PEP	PepsiCO Inc	3020	5410
40	PFE	Pfizer Inc	3520	5620
41	PG	Procter&Gamble Co	3030	5420
42	S	Sprint Corp	5010	5810
43	SLB	Schlumberger Ltd	1010	5010
44	SO	Southern Co	5510	5910
45	T	AT&T Inc	5010	5810
46	UTX	United Technologies Corp	2010	5210
47	VZ	Verizon Communications Inc	5010	5810
48	WFC	Wells Fargo&Co	4010	5510
49	WMT	Wal-Mart Stores Inc	3010	5430
50	XOM	ExxonMobil Corp	1010	5010
51	GS	Goldman Sachs Group Inc	4020	5510
52	IBP.1	IBP Inc	3020	5220
53	V	Visa Inc	4510	5720
54	AMZN	Amazon.com Inc	2550	5340

The whole vine structure spans nodes in Fig. 4.8 is:
(40, 0)(40, 2)(33, 3)(25, 1)(9, 2)(47, 40)(44, 3)(37, 36)(40, 3)(39, 0)(30, 3)(37, 6)(47, 6)(10, 9)(25, 5)(35, 18)(33, 7)(42, 20)(18, 0)(54, 5)(37, 34)(38, 10)(28, 3)(40, 29)(49, 2)(45, 9)(19, 6)(24, 6)(41, 40)(40, 22)(38, 17)(13, 6)(35, 16)(48, 6)(46, 22)(37, 23)(25, 0)(20, 6)(49, 14)(51, 0)(50, 3)(48, 26)(34, 21)(52, 47)(6, 4)(10, 8)(38, 12)(11, 7)(15, 10)(32, 19)(43, 2)(27, 5)(53, 34)(31, 3)(40, 9)(2, 0)(40, 6)(40, 18)(33, 30)(40, 30)(47, 20)(40, 28)(45, 2)(22, 0)(29, 0)(36, 6)(47, 3)(34, 23)(23, 6)(47, 37)(47, 22)(5, 0)(7, 3)(37, 13)(24, 19)(38, 9)(37, 24)(47, 4)(45, 10)(5, 1)(40, 25)(48, 13)(41, 22)(50, 30)(44, 28)(26, 6)(9, 8)(39, 25)(35, 0)(51, 25)(42, 6)(27, 25)(46, 40)(49, 9)(33, 11)(15, 8)(12, 10)(52, 40)(54, 27)(40, 31)(53, 37)(14, 2)(37, 21)(18, 16)(17, 12)(32, 6)(43, 40)(40, 39)(40, 37)(17, 10)(18, 2)(22,

18)(30, 7)(47, 36)(47, 0)(39, 5)(6, 3)(47, 42)(49, 40)(9, 0)(30, 28)(47, 23)(27, 1)(34, 6)(47, 24)(40, 33)(22, 6)(29, 2)(37, 19)(45, 40)(51, 40)(40, 35)(47, 28)(48, 37)(29, 25)(47, 41)(10, 2)(37, 20)(24, 13)(16, 0)(47, 46)(50, 33)(40, 4)(45, 8)(1, 0)(23, 21)(38, 8)(52, 3)(53, 23)(32, 24)(11, 3)(12, 9)(14, 9)(43, 0)(15, 9)(31, 28)(26, 13)(54, 25)(44, 40)(40, 10)(6, 0)(47, 18)(47, 44)(18, 9)(41, 6)(40, 36)(37, 3)(40, 5)(45, 0)(40, 23)(40, 24)(47, 19)(22, 2)(36, 20)(39, 29)(47, 34)(17, 9)(44, 30)(49, 0)(40, 14)(50, 7)(29, 18)(28, 6)(51, 29)(35, 2)(40, 16)(33, 28)(21, 6)(46, 41)(48, 24)(47, 13)(37, 22)(37, 26)(25, 2)(37, 4)(45, 15)(52, 6)(54, 1)(12, 8)(50, 40)(8, 2)(27, 0)(43, 18)(37, 32)(53, 6)(42, 37)(38, 15)(44, 31)(39, 1)(30, 11)(37, 0)(18, 6)(10, 0)(24, 22)(36, 3)(39, 2)(29, 5)(37, 28)(23, 3)(47, 2)(44, 33)(40, 13)(24, 23)(35, 9)(47, 30)(48, 47)(50, 28)(40, 1)(17, 8)(40, 7)(41, 0)(29, 9)(36, 4)(47, 32)(51, 2)(40, 20)(19, 13)(16, 2)(40, 8)(29, 22)(47, 21)(43, 22)(14, 0)(46, 6)(54, 0)(45, 38)(15, 2)(25, 18)(52, 28)(42, 36)(26, 24)(45, 18)(53, 21)(49, 18)(39, 27)(15, 12)(31, 30)(50, 11)(44, 6)(40, 34)(30, 6)(44, 37)(40, 19)(24, 0)(39, 18)(37, 18)(6, 2)(22, 9)(36, 28)(34, 3)(28, 7)(47, 33)(5, 2)(47, 29)(34, 24)(38, 2)(22, 13)(18, 10)(23, 22)(20, 3)(52, 44)(54, 39)(16, 9)(49, 35)(25, 22)(46, 0)(48, 19)(40, 21)(49, 29)(50, 44)(51, 18)(28, 23)(41, 37)(20, 4)(8, 0)(40, 11)(40, 15)(45, 35)(18, 14)(42, 40)(17, 15)(47, 43)(32, 13)(40, 27)(45, 12)(47, 31)(53, 47)(29, 1)(47, 26)(37, 2)(35, 10)(48, 40)(40, 38)(37, 30)(41, 24)(54, 40)(23, 0)(33, 6)(24, 21)(34, 22)(49, 22)(26, 19)(44, 7)(50, 47)(18, 5)(29, 14)(47, 25)(51, 22)(29, 6)(40, 32)(29, 27)(45, 16)(24, 18)(18, 8)(39, 22)(46, 37)(45, 17)(49, 16)(24, 3)(15, 0)(44, 23)(36, 23)(2, 1)(22, 19)(53, 40)(12, 2)(25, 9)(23, 13)(52, 37)(35, 29)(4, 3)(28, 20)(34, 28)(28, 11)(43, 29)(42, 3)(31, 6)(47, 9)(44, 34)(30, 23)(47, 39)(47, 7)(34, 13)(23, 19)(16, 10)(23, 18)(37, 29)(38, 0)(34, 0)(40, 12)(22, 5)(35, 8)(22, 3)(36, 34)(28, 24)(24, 2)(37, 33)(51, 9)(54, 29)(25, 6)(17, 2)(44, 11)(23, 20)(35, 22)(53, 24)(41, 18)(50, 6)(22, 21)(35, 14)(46, 24)(43, 25)(48, 22)(27, 2)(40, 26)(18, 1)(52, 23)(49, 25)(28, 4)(18, 15)(42, 28)(37, 31)(49, 45)(32, 22)(29, 16)(3, 0)(39, 6)(9, 6)(44, 36)(46, 18)(21, 0)(34, 30)(34, 19)(36, 24)(13, 3)(37, 25)(40, 17)(34, 20)(29, 24)(47, 5)(47, 11)(33, 23)(23, 2)(51, 49)(22, 1)(23, 4)(22, 16)(49, 10)(16, 8)(51, 47)(32, 23)(48, 23)(53, 22)(54, 2)(35, 25)(42, 4)(50, 37)(16, 14)(41, 23)(52, 30)(35, 15)(28, 22)(45, 29)(27, 18)(26, 22)(43, 6)(31, 23)(7, 6)(34, 18)(12, 0)(38, 18)(37, 9)(18, 3)(44, 20)(37, 7)(39, 37)(24, 20)(34, 2)(13, 0)(6, 5)(28, 0)(51, 6)(19, 3)(29, 23)(18, 12)(45, 22)(48, 34)(41, 2)(38, 35)(29, 10)(21, 18)(34, 33)(25, 24)(36, 22)(34, 4)(51, 35)(26, 23)(50, 23)(53, 0)(16, 15)(49, 47)(43, 37)(49, 8)(54, 18)(25, 16)(34, 32)(52, 33)(22, 14)(11, 6)(34, 31)(42, 23)(27, 22)(36, 30)(17, 0)(46, 23)(47, 1)(36, 33)(36, 0)(3, 2)(47, 35)(24, 9)(29, 8)(35, 12)(34, 29)(37, 5)(18, 13)(34, 26)(51, 37)(39, 9)(25, 23)(50, 34)(19, 0)(44, 24)(23, 7)(21, 2)(44, 4)(47, 27)(28, 13)(49, 6)(52, 50)(48, 3)(41, 29)(42, 34)(38, 16)(32, 3)(6, 1)(30, 20)(46, 2)(51, 16)(54, 22)(25, 14)(18, 17)(45, 25)(53, 18)(49, 15)(43, 9)(37, 11)(33, 31)(22, 20)(22, 10)(28, 19)(44, 22)(30, 4)(33, 20)(23, 9)(13, 2)(49, 38)(54, 47)(29, 3)(20, 0)(35, 6)(25, 10)(22, 8)(34, 7)(51, 24)(30, 24)(47, 16)(9, 5)(16, 12)(37, 1)(34, 25)(39, 24)(44, 42)(19, 18)(49, 37)(50, 36)(35, 17)(41, 34)(36, 13)(50, 31)(51, 45)(27, 6)(29, 15)(29, 21)(26, 3)(32, 0)(53, 2)(52, 34)(46, 29)(51, 14)(23, 11)(43, 24)(48, 32)(47, 45)(44, 0)(37, 35)(54, 6)(39, 23)(24, 5)(51, 23)(33, 24)(34, 9)(48, 0)(52, 7)(25, 3)(16, 6)(19, 2)(49, 24)(38, 29)(20, 13)(51, 10)(36, 7)(22, 15)(29, 13)(47, 14)(33, 4)(49, 12)(36, 19)(30, 22)(32,

28)(17, 16)(41, 25)(9, 1)(28, 18)(25, 8)(21, 3)(37, 27)(53, 29)(34, 11)(43, 23)(32, 26)(52, 31)(42, 30)(46, 34)(50, 20)(30, 0)(9, 3)(47, 10)(44, 13)(51, 34)(45, 6)(41, 9)(49, 17)(23, 5)(28, 2)(49, 23)(21, 13)(36, 32)(43, 34)(36, 18)(20, 19)(54, 37)(39, 34)(37, 16)(50, 4)(26, 0)(20, 7)(25, 15)(52, 11)(53, 3)(14, 6)(38, 22)(46, 25)(25, 21)(29, 19)(24, 1)(27, 9)(51, 8)(42, 33)(29, 12)(48, 28)(36, 11)(50, 24)(31, 7)(33, 22)(35, 24)(47, 8)(24, 7)(45, 37)(29, 28)(35, 23)(10, 6)(46, 9)(33, 0)(7, 4)(29, 17)(51, 3)(39, 3)(36, 2)(20, 18)(44, 19)(41, 3)(54, 9)(20, 11)(21, 19)(22, 12)(38, 25)(48, 36)(30, 13)(50, 22)(37, 14)(43, 3)(49, 34)(28, 26)(32, 20)(53, 13)(21, 9)(51, 15)(25, 13)(23, 1)(24, 16)(52, 36)(27, 24)(31, 11)(50, 42)(34, 5)(44, 18)(35, 34)(13, 9)(37, 10)(48, 20)(20, 2)(22, 17)(30, 19)(45, 24)(34, 1)(24, 14)(24, 11)(49, 3)(22, 7)(46, 3)(5, 3)(8, 6)(25, 19)(36, 29)(43, 21)(24, 4)(51, 21)(51, 39)(23, 16)(41, 39)(36, 26)(47, 15)(52, 20)(53, 19)(51, 38)(44, 32)(25, 12)(50, 0)(42, 7)(28, 21)(27, 23)(36, 31)(54, 24)(33, 13)(30, 18)(48, 44)(35, 3)(44, 2)(51, 5)(19, 9)(42, 24)(3, 1)(37, 8)(49, 39)(47, 38)(28, 25)(41, 5)(24, 10)(7, 0)(29, 20)(25, 17)(46, 39)(45, 23)(22, 4)(39, 21)(51, 12)(34, 16)(33, 19)(34, 27)(52, 24)(11, 4)(53, 25)(54, 23)(23, 14)(43, 13)(15, 6)(32, 18)(36, 21)(50, 13)(51, 43)(31, 20)(26, 20)(36, 25)(16, 3)(28, 9)(30, 2)(23, 10)(39, 35)(42, 22)(21, 5)(33, 18)(47, 12)(38, 6)(49, 5)(27, 3)(44, 29)(24, 8)(48, 18)(42, 11)(44, 26)(45, 34)(4, 0)(51, 41)(51, 17)(41, 1)(21, 20)(51, 13)(32, 2)(46, 5)(54, 34)(50, 19)(53, 9)(31, 24)(34, 14)(37, 15)(52, 4)(43, 19)(43, 39)(13, 7)(19, 7)(30, 29)(38, 37)(48, 2)(23, 8)(39, 13)(34, 10)(39, 16)(49, 41)(36, 9)(51, 19)(47, 17)(45, 3)(33, 2)(54, 3)(35, 5)(44, 21)(26, 18)(51, 46)(14, 3)(49, 21)(51, 1)(13, 4)(22, 11)(32, 29)(50, 18)(42, 0)(12, 6)(43, 5)(41, 27)(24, 15)(52, 42)(53, 43)(53, 28)(25, 20)(31, 4)(10, 3)(19, 4)(54, 41)(30, 21)(18, 7)(17, 6)(39, 14)(44, 25)(33, 29)(45, 39)(52, 22)(34, 8)(53, 51)(41, 35)(13, 5)(37, 12)(39, 19)(38, 24)(11, 0)(48, 29)(42, 13)(51, 27)(26, 2)(23, 15)(41, 21)(16, 5)(46, 1)(49, 1)(53, 36)(32, 30)(49, 43)(50, 2)(43, 28)(20, 9)(42, 31)(44, 9)(39, 10)(37, 17)(51, 28)(38, 23)(19, 5)(18, 4)(33, 21)(42, 19)(8, 3)(14, 5)(35, 21)(7, 2)(53, 20)(50, 29)(33, 32)(52, 0)(41, 16)(29, 26)(49, 13)(43, 36)(34, 15)(45, 5)(48, 30)(54, 51)(49, 46)(49, 27)(21, 1)(31, 22)(13, 11)(43, 41)(53, 39)(30, 25)(24, 12)(39, 28)(49, 19)(30, 9)(24, 17)(23, 12)(15, 3)(10, 5)(51, 36)(43, 20)(39, 8)(33, 25)(45, 41)(42, 18)(41, 13)(41, 14)(30, 26)(29, 7)(50, 32)(53, 44)(48, 33)(19, 11)(54, 49)(50, 21)(4, 2)(53, 5)(21, 16)(31, 0)(46, 21)(35, 1)(52, 13)(43, 35)(27, 21)(38, 34)(38, 3)(28, 5)(21, 7)(39, 36)(41, 19)(23, 17)(41, 10)(34, 12)(33, 9)(50, 25)(51, 20)(43, 1)(46, 35)(35, 13)(8, 5)(42, 2)(32, 7)(29, 4)(53, 30)(33, 26)(45, 21)(18, 11)(21, 14)(16, 1)(52, 19)(31, 13)(53, 49)(44, 43)(54, 21)(35, 27)(39, 15)(50, 48)(36, 5)(21, 4)(49, 28)(51, 44)(35, 19)(39, 38)(34, 17)(12, 3)(52, 18)(46, 16)(39, 20)(32, 4)(48, 7)(41, 8)(31, 19)(11, 2)(50, 9)(25, 7)(53, 33)(14, 1)(42, 29)(13, 1)(21, 10)(15, 5)(43, 27)(53, 41)(27, 16)(54, 35)(45, 14)(43, 30)(50, 26)(9, 7)(44, 39)(41, 28)(25, 4)(38, 5)(17, 3)(20, 5)(43, 33)(51, 30)(52, 2)(19, 1)(48, 4)(49, 36)(39, 12)(31, 18)(53, 50)(42, 21)(42, 32)(41, 15)(27, 13)(14, 10)(46, 14)(54, 16)(21, 8)(45, 1)(53, 35)(54, 43)(27, 14)(26, 7)(29, 11)(39, 30)(26, 4)(48, 42)(35, 28)(46, 45)(21, 11)(41, 36)(44, 5)(39, 17)(54, 13)(53, 1)(51, 33)(10, 1)(41, 38)(14, 8)(52, 29)(54, 14)(21, 15)(9, 4)(32, 11)(45, 27)(27, 19)(50, 43)(49, 20)(53, 7)(12, 5)(42, 25)(43, 16)(31, 2)(30, 5)(54, 19)(42, 26)(28, 1)(36, 35)(42, 9)(17, 5)(46, 10)(41, 20)(27, 10)(38, 21)(49, 44)(53, 4)(51, 50)(54, 45)(16, 13)(31, 29)(8, 1)(43, 7)(52, 21)(25, 11)(52, 32)(41, 12)(43, 14)(15,

14)(53, 27)(48, 11)(39, 33)(36, 1)(33, 5)(49, 30)(28, 27)(41, 17)(38, 14)(21, 12)(44, 41)(15, 1)(46, 8)(11, 9)(19, 16)(54, 10)(14, 13)(26, 11)(51, 7)(53, 42)(27, 8)(50, 39)(32, 31)(54, 53)(52, 48)(52, 25)(43, 4)(35, 20)(32, 21)(45, 43)(54, 28)(48, 21)(52, 9)(44, 35)(45, 13)(20, 1)(49, 33)(32, 25)(51, 4)(36, 27)(14, 12)(39, 7)(19, 14)(50, 5)(38, 1)(52, 26)(46, 15)(21, 17)(54, 8)(53, 11)(27, 15)(53, 16)(43, 42)(31, 21)(43, 10)(41, 30)(35, 30)(26, 21)(28, 16)(17, 14)(39, 4)(38, 27)(13, 10)(27, 20)(7, 5)(45, 19)(44, 1)(41, 33)(50, 49)(32, 9)(48, 25)(51, 42)(43, 11)(53, 52)(12, 1)(43, 8)(46, 27)(48, 31)(54, 15)(54, 36)(53, 14)(5, 4)(48, 9)(19, 10)(54, 38)(54, 20)(49, 7)(17, 1)(30, 1)(36, 16)(31, 25)(26, 25)(50, 41)(13, 8)(53, 32)(46, 38)(42, 39)(35, 33)(44, 27)(51, 11)(43, 15)(53, 45)(27, 12)(52, 43)(28, 14)(33, 1)(50, 35)(41, 7)(19, 8)(45, 28)(31, 9)(46, 12)(36, 14)(54, 44)(20, 16)(53, 48)(27, 17)(52, 51)(43, 32)(30, 27)(15, 13)(26, 9)(49, 4)(54, 46)(39, 11)(43, 38)(53, 10)(42, 5)(28, 10)(50, 1)(44, 16)(35, 7)(38, 13)(54, 30)(45, 36)(46, 17)(33, 27)(53, 26)(11, 5)(41, 4)(53, 8)(52, 39)(49, 42)(46, 43)(53, 31)(51, 32)(20, 14)(19, 15)(54, 12)(48, 43)(38, 19)(51, 48)(45, 20)(35, 4)(54, 33)(54, 17)(36, 10)(30, 16)(42, 41)(50, 27)(43, 12)(28, 8)(43, 26)(44, 14)(49, 11)(7, 1)(46, 13)(39, 32)(52, 5)(53, 15)(31, 26)(45, 44)(30, 14)(32, 5)(46, 19)(36, 8)(4, 1)(33, 16)(48, 39)(13, 12)(20, 10)(52, 49)(54, 50)(42, 35)(27, 7)(53, 38)(43, 17)(41, 11)(28, 15)(43, 31)(51, 26)(44, 10)(20, 8)(19, 12)(52, 41)(50, 16)(17, 13)(33, 14)(49, 32)(39, 26)(45, 30)(54, 7)(48, 5)(53, 46)(35, 11)(51, 31)(42, 1)(36, 15)(38, 28)(27, 4)(30, 10)(26, 5)(41, 32)(54, 4)(44, 8)(16, 7)(52, 35)(49, 48)(46, 28)(38, 36)(39, 31)(20, 15)(45, 33)(50, 14)(42, 27)(53, 12)(11, 1)(19, 17)(54, 42)(31, 5)(30, 8)(44, 15)(48, 41)(46, 36)(16, 4)(38, 20)(28, 12)(53, 17)(35, 32)(52, 1)(50, 45)(49, 26)(27, 11)(14, 7)(33, 10)(28, 17)(44, 38)(50, 10)(45, 7)(32, 1)(41, 26)(48, 35)(46, 20)(33, 8)(42, 16)(54, 11)(14, 4)(30, 15)(52, 27)(36, 12)(49, 31)(38, 30)(46, 44)(36, 17)(50, 8)(10, 7)(33, 15)(35, 26)(45, 4)(54, 52)(42, 14)(20, 12)(48, 1)(16, 11)(41, 31)(32, 27)(44, 12)(38, 33)(8, 7)(50, 15)(10, 4)(45, 42)(46, 30)(20, 17)(14, 11)(35, 31)(26, 1)(48, 27)(52, 16)(54, 32)(50, 38)(44, 17)(8, 4)(45, 11)(31, 1)(30, 12)(54, 48)(42, 10)(52, 14)(32, 16)(27, 26)(15, 7)(46, 33)(38, 7)(30, 17)(52, 45)(54, 26)(42, 8)(33, 12)(32, 14)(31, 27)(50, 46)(48, 16)(15, 4)(11, 10)(46, 7)(33, 17)(50, 12)(38, 4)(52, 10)(11, 8)(48, 14)(45, 32)(26, 16)(54, 31)(42, 15)(50, 17)(12, 7)(52, 8)(46, 4)(31, 16)(32, 10)(26, 14)(15, 11)(42, 38)(48, 45)(17, 7)(32, 8)(12, 4)(48, 10)(46, 42)(45, 26)(38, 11)(31, 14)(52, 15)(52, 38)(17, 4)(46, 11)(48, 8)(26, 10)(32, 15)(45, 31)(42, 12)(52, 46)(31, 10)(38, 32)(26, 8)(48, 15)(42, 17)(12, 11)(48, 38)(52, 12)(26, 15)(31, 8)(46, 32)(17, 11)(32, 12)(31, 15)(38, 26)(48, 46)(52, 17)(38, 31)(48, 12)(46, 26)(32, 17)(26, 12)(46, 31)(48, 17)(26, 17)(31, 12)(31, 17)

Appendix B
Data Acquisition

Data acquisition from StockTwits is described as follows.

The StockTwits API provides snippets of posts in a JSON file at the URL address: https://api.stocktwits.com/api/2/streams/symbol/ticker_name.json

The file structure contains posts with associated information such as post_ID, text, timestamp, user, source, sentiment, etc. A sample is provided as below, where sensitive data is anonymized with "?."

```
{"response":{"status":???},
/*one post*/
{"id":1243???21,
"body":"$A??? Another green Premarket. Lets see how the day
    goes.",
"created_at":"2018-0?-??T08:??:??Z",
"user":{"id":13???41,"username":"Mr???h","name":"Wil???k","
    avatar_url":"http://avatars.sto???.jpg",
"avatar_url_ssl":"https://s3.ama???/images/???mb.jpg","
    join_date":"2017-??-?1","official":false,"identity":"User
    ","classification":[],"followers":?,"following":?,"ideas
    ":4?1,"watchlist_stocks_count":1?,"like_count":1?1},
"source":{"id":2???,"title":"St???oid ","url":"http://www.???
    le"},
"symbols":[{"id":???,"symbol":"A???","title":"A???","aliases
    ":[],"is_following":false,"watchlist_count":2???73}],
"mentioned_users":[],"
entities":{"sentiment":{"basic":"Bullish"}
},
/*another post*/
...
```


F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4

The URLs are frequently requested with the “urllib” Python package. The file is stored in a temporary variable `datatmp` and loaded with a JSON parser. After all the useful name/value pairs are recorded to a CSV file, the temporary variable is emptied for reuse.

```
    reader = codecs.getreader("utf-8")
    req = urllib.request.Request(url)
    req.add_header('Pragma', 'no-cache')
    datatmp = urllib.request.build_opener().open(req)

    data = json.load(reader(datatmp))
    del datatmp
    urllib.request.urlcleanup()
```

References

1. K. Aas, D. Berg, Models for construction of multivariate dependence – a comparison study. Eur. J. Financ. **15**, 639–659 (2009)
2. A. Ang, *Asset Management: A Systematic Approach to Factor Investing* (Oxford University Press, Oxford, 2014)
3. W. Antweiler, M.Z. Frank, Is all that talk just noise? The information content of internet stock message boards. J. Financ. **59**(3), 1259–1294 (2004)
4. D. Avramov, G. Zhou, Bayesian portfolio analysis. Annu. Rev. Financ. Econ. **2**, 25–47 (2010)
5. S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining, in *7th Language Resources and Evaluation Conference*, 2010, pp. 2200–2204
6. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in *International Conference on Learning Representations (ICLR)*, 2015, pp. 1–15
7. H. Bai, F.Z. Xing, E. Cambria, W.-B. Huang, Business taxonomy construction using concept-level hierarchical clustering, in *The First Workshop on Financial Technology and Natural Language Processing (FinNLP-IJCAI)*, 2019, pp. 1–7
8. S. Banerjee, R. Kaniel, I. Kremer, Price drift as an outcome of differences in higher-order beliefs. Rev. Financ. Stud. **22**(9), 3707–3734 (2009)
9. T. Bao, C. Diks, H. Li, A generalized capm model with asymmetric power distributed errors with an application to portfolio construction. Econ. Model. **68**, 611–621 (2018)
10. T. Bedford, R.M. Cooke, Probability density decomposition for conditionally dependent random variables modeled by vines. Ann. Math. Artif. Intell. **32**, 245–268 (2001)
11. T. Bedford, R.M. Cooke, Vines: a new graphical model for dependent random variables. Ann. Stat. **30**(4), 1031–1068 (2002)
12. Y. Bengio, R. Ducharme, P. Vincent, A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
13. M.J. Best, J. Hlouskova, An algorithm for portfolio optimization with transaction costs. Manag. Sci. **51**(11), 1676–1688 (2005)
14. F. Black, R. Litterman, Asset allocation: combining investor view with market equilibrium. J. Fixed Income **1**, 7–18 (1991)
15. D.M. Blei, Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2012)
16. J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007, pp. 440–447
17. D. Bolinger, *Aspects of Language* (Harcourt Brace Jovanovich, New York, 1975)

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4

18. J. Bollen, H. Mao, A. Pepe, Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena, in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011
19. J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market. J. Comput. Sci. **2**(1), 1–8 (2011)
20. A. Bordes, Y.-L. Boureau, J. Weston, Learning end-to-end goal-oriented dialog, in *International Conference on Learning Representations (ICLR)*, 2017, pp. 1–15
21. A. Bordes, S. Ertekin, J. Weston, L. Bottou, Fast kernel classifiers with online and active learning. J. Mach. Learn. Res. **6**, 1579–1619 (2005)
22. A. Brabazon, M. O'Neill, An introduction to evolutionary computation in finance. IEEE Comput. Intell. Mag. **3**(4), 42–55 (2008)
23. R.J. Brachman, T. Khabaza, W. Kloesgen, G. Piatetsky-Shapiro, E. Simoudis, Mining business databases. Commun. ACM **39**(11), 42–48 (1996)
24. E. Cambria, An introduction to concept-level sentiment analysis, in *Mexican International Conference on Artificial Intelligence (LNCS)*, vol. 8266, 2013, pp. 478–483
25. E. Cambria, Affective computing and sentiment analysis. IEEE Intell. Syst. **31**(2), 102–107 (2016)
26. E. Cambria, A. Hussain, *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis* (Springer International Publishing, Cham, 2015)
27. E. Cambria, A. Livingstone, A. Hussain, The hourglass of emotions, in *Lecture Notes in Computer Science*, vol. 7403 (Springer, Berlin, 2012), pp. 144–157
28. E. Cambria, S. Poria, D. Hazarika, K. Kwok, Senticnet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings, in *The Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 1795–1802
29. E. Cambria, D. Rajagopal, K. Kwok, J. Sepulveda, Gecka: game engine for commonsense knowledge acquisition, in *The Twenty-Eighth International Flairs Conference*, 2015, pp. 282–287
30. E. Cambria, B. White, Jumping NLP curves: a review of natural language processing research. IEEE Comput. Intell. Mag. **9**(2), 48–57 (2014)
31. L.K.C. Chan, J. Lakonishok, B. Swaminathan, Industry classification and return comovement. Financ. Anal. J. **63**(6), 56–70 (2007)
32. S.W. Chan, J. Franklin, A text-based decision support system for financial sequence prediction. Decis. Support. Syst. **52**(1), 189–198 (2011)
33. I. Chaturvedi, Y.-S. Ong, I. Tsang, R.E. Welsch, E. Cambria, Learning word dependencies in text by means of a deep recurrent belief network. Knowl. Based Syst. **108**, 144–154 (2016)
34. H. Chen, X. Liu, D. Yin, J. Tang, A survey on dialogue systems: recent advances and new frontiers. ACM SIGKDD Explor. Newslett. **19**(2), 25–35 (2017)
35. H. Choi, H. Varian, Predicting the present with Google trends. Econ. Rec. **88**(1), 2–9 (2012)
36. Y. Choi, C. Cardie, Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification, in *Empirical Methods in Natural Language Processing (EMNLP)*, 2009, pp. 590–598
37. N. Chomsky, Three models for the description of language. IRE Trans. Inf. Theory **2**(3), 113–124 (1956)
38. G.P.E. Clarkson, A model of the trust investment process, in *Computers and Thought* (McGraw-Hill, New York, 1963), pp. 347–374
39. R.M. Cooke, D. Kurowicka, K. Wilson, Sampling, conditionalizing, counting, merging, searching regular vines. J. Multivar. Anal. **138**, 4–18 (2015)
40. W. Croft, D.A. Cruse, *Cognitive Linguistics* (Cambridge University Press, New York, 2004)
41. Z. Da, Q. Liu, E. Schaumburg, Decomposing short-term return reversal. Technical Report Staff Report no. 513, Federal Reserve Bank of New York, 2011
42. S.R. Das, M.Y. Chen, Yahoo! for Amazon: sentiment extraction from small talk on the web. Manag. Sci. **53**(9), 1375–1388 (2007)
43. A.B. Davidow, J.D. Peterson, A modern approach to asset allocation and portfolio construction. Technical Report MKT81752HL-02, Schwab Center for Financial Research, 2014

44. M.-Y. Day, T.-K. Cheng, J.-G. Li, Ai robo-advisor with big data analytics for financial services, in *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018, pp. 1027–1031
45. M.-Y. Day, J.-T. Lin, Y.-C. Chen, Artificial intelligence for conversational robo-advisor, in *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018, pp. 1057–1064
46. M.-C. de Marneffe, C.D. Manning, The Stanford typed dependencies representation, in *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 2008, pp. 1–8
47. Deloitte, Robo-advisory in wealth management. Communication of Deloitte GmbH, 2016
48. R. Dey, F.M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, in *IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017, pp. 1597–1600
49. X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction, in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015
50. W. Du, S. Tan, X. Cheng, X. Yun, Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon, in *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM)*, 2010, pp. 111–120
51. F. Durante, C. Sempi, *Principles of Copula Theory* (CRC Press, Boca Raton, 2016)
52. G. Elidan, Copulas in machine learning, in *Copulae in Mathematical and Quantitative Finance*, vol. 213 (Springer, Berlin/Heidelberg, 2013), pp. 39–60
53. I. Evstigneev, T. Hens, K.R. Schenk-Hoppe (eds.), *Mathematical Financial Economics: A Basic Introduction* (Springer, New York, 2015)
54. E.F. Fama, Efficient capital markets: a review of theory and empirical work. J. Financ. **25**, 383–417 (1970)
55. E.F. Fama, K.R. French, Luck versus skill in the cross-section of mutual fund returns. J. Financ. **65**(5), 1915–1947 (2010)
56. E.F. Fama, K.R. French, A five-factor asset pricing model. J. Financ. Econ. **116**, 1–22 (2015)
57. F. Fancellu, A. Lopez, B. Webber, H. He, Detecting negation scope is easy, except when it isn't, in *European Chapter of the Association for Computational Linguistics (EACL)*, 2017, pp. 58–63
58. R. Feldman, Techniques and applications for sentiment analysis. Commun. ACM **56**(4), 82–89 (2013)
59. A. Felfernig, K. Isak, K. Szabo, P. Zachar, The vita financial services sales support environment, in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, 2007, pp. 1692–1699
60. C. Fellbaum, *WordNet: An Electronic Lexical Database* (MIT Press, Cambridge, 1998)
61. K.B. Frazier, R.W. Ingram, B.M. Tennyson, A methodology for the analysis of narrative accounting disclosures. J. Account. Res. **22**(1), 318–331 (1984)
62. G.P.C. Fung, J.X. Yu, W. Lam, Stock prediction: integrating text mining approach using real-time news, in *2003 IEEE International Conference on Computational Intelligence for Financial Engineering, Proceedings*, 2003, pp. 395–402
63. P. Fung, A. Dey, F.B. Siddique, R. Lin, Y. Yang, D. Bertero, Y. Wan, R.H.Y. Chan, C.-S. Wu, Zara: a virtual interactive dialogue system incorporating emotion, sentiment and personality recognition, in *International Conference on Computational Linguistics (COLING)*, 2016, pp. 278–281
64. J. Gagnon, S. Goyal, Networks, markets, and inequality. Am. Econ. Rev. **107**(1), 1–30 (2017)
65. F.A. Gers, D. Eck, J. Schmidhuber, *Applying LSTM to Time Series Predictable Through Time-Window Approaches* (Springer, London, 2002), pp. 193–200
66. C.W.J. Granger, Investigating causal relations by econometric models and cross-spectral methods. Econometrica **37**(3), 424–438 (1969)
67. K. Greff, R.K. Srivastava, J. Koutnik, B.R. Steunebrink, J. Schmidhuber, LSTM: a search space odyssey. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2222–2232 (2017)

68. S.S. Groth, J. Muntermann, An intraday market risk management approach based on textual analysis. Decis. Support. Syst. **50**(4), 680–691 (2011)
69. R.V. Guha, D.B. Lenat, Cyc: a midterm report. AI Mag. **11**(3), 32–59 (1990)
70. W.L. Hamilton, K. Clark, J. Leskovec, D. Jurafsky, Inducing domain-specific sentiment lexicons from unlabeled corpora, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 595–605
71. J. Hawley, J. Lukomnik, The purpose of asset management. Technical report, Pension Insurance Corporation, 2018
72. G. He, R. Litterman, The intuition behind black-Litterman model portfolios. Goldman Sachs working paper (1999). https://doi.org/10.2139/ssrn.334304
73. E. Henry, Are investors influenced by how earnings press releases are written? Int. J. Bus. Commun. **45**, 363–407 (2008)
74. M. Hentschel, O. Alonso, Follow the money: a study of cashtags on Twitter. First Monday **19**(8) (2014). https://doi.org/10.5210/fm.v19i8.5385
75. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
76. M. Hu, B. Liu, Mining and summarizing customer reviews, in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 168–177
77. J. Huang, Taxable and tax-deferred investing: a tax-arbitrage approach. Rev. Financ. Stud. **21**(5), 2173–2207 (2008)
78. K.K. Hung, C.C. Cheung, L. Xu, New Sharpe-ratio-related methods for portfolio selection, in *Proceedings of the Conference on Computational Intelligence for Financial Engineering (CIFEr)*, 2000, pp. 34–37
79. R.J. Hyndman, A.B. Koehler, Another look at measures of forecast accuracy. Int. J. Forecast. **22**(4), 679–688 (2006)
80. R. Jensen, The digital provide: information (technology), market performance, and welfare in the south Indian fisheries sector. Q. J. Econ. **122**(3), 879–924 (2007)
81. D. Jung, V. Dorner, F. Glaser, S. Morana, Robo-advisory: digitalization and automation of financial advisory. Bus. Inf. Syst. Eng. **60**(1), 81–86 (2018)
82. D. Jung, V. Dorner, C. Weinhardt, H. Pusmaz, Designing a robo-advisor for risk-averse, low-budget consumers. Electron. Mark. **28**, 367–380 (2018)
83. N.K. Kasabov, Q. Song, Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Trans. Fuzzy Syst. **10**, 144–154 (2002)
84. E.F. Kelly, *Computer Recognition of English Word Senses* (North-Holland Publishing Co, Amsterdam, 1975)
85. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in *Proceedings of International Conference on Learning Representations*, 2015
86. B.S. Kumar, V. Ravi, A survey of the applications of text mining in financial domain. Knowl.-Based Syst. **114**, 128–147 (2016)
87. D. Kurowicka, H. Joe (eds.), *Dependence Modeling: Vine Copula Handbook* (World Scientific, London, 2011)
88. V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, J. Allan, Language models for financial news recommendation, in *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM)*, 2000, pp. 389–396
89. Q. Le, T. Mikolov, Distributed representations of sentences and documents, in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014, pp. 1188–1196
90. J.K. Lee, R.R. Trippi, S.C. Chu, H.S. Kim, K-folio: integrating the Markowitz model with a knowledge-based system. J. Portf. Manag. **17**(1), 89–93 (1990)
91. G. Leech, *Semantics: The Study of Meaning*, 2 edn. (Harmondsworth, Penguin, 1981)
92. B. Lemaire, G. Denhiere, C. Bellissens, S. Jhean-Larose, A computational model for simulating text comprehension. Behav. Res. Methods **38**(4), 628–637 (2006)
93. O. Levy, Y. Goldberg, Dependency-based word embeddings, in *The 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014, pp. 302–308

94. L. Li, B. Qin, W. Ren, T. Liu, Truth discovery with memory network. Tsinghua Sci. Technol. **22**(6), 609–618 (2017)
95. Q. Li, L. Jiang, P. Li, H. Chen, Tensor-based learning for predicting stock movements, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, 2015, pp. 1784–1790
96. Q. Li, T. Wang, P. Li, L. Liu, Q. Gong, Y. Chen, The effect of news and public mood on stock movements. Inf. Sci. **278**, 826–840 (2014)
97. X. Li, H. Xie, Y. Song, S. Zhu, Q. Li, F.L. Wang, Does summarization help stock prediction? A news impact analysis. IEEE Intell. Syst. **30**(3), 26–34 (2015)
98. B. Liu, Many facets of sentiment analysis, in *A Practical Guide to Sentiment Analysis* (Springer, Cham, 2017), pp. 11–39
99. C. Liu, S.C.H. Hoi, P. Zhao, J. Sun, Online Arima algorithms for time series prediction, in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016
100. H. Liu, P. Singh, Conceptnet – a practical commonsense reasoning tool-kit. BT Technol. J. **22**(4), 211–226 (2004)
101. G.M. Ljung, G.E.P. Box, On a measure of lack of fit in time series models. Biometrika **65**(2), 297–303 (1978)
102. A.W. Lo, *Adaptive Markets: Financial Evolution at the Speed of Thought* (Princeton University Press, Princeton, 2017)
103. T. Loughran, B. McDonald, When is a liability not a liability? Textual analysis, dictionaries, and 10-ks. J. Financ. **66**, 67–97 (2011)
104. L. Luo, Y. Xiong, Y. Liu, X. Sun, Adaptive gradient methods with dynamic bound of learning rate, in *Proceedings of International Conference on Learning Representations*, 2019
105. H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM)*, 2011, pp. 287–296
106. S. Makridakis, M. Hibon, The M3-competition: results, conclusions and implications. Int. J. Forecast. **16**(4), 451–476 (2000)
107. A.S. Manek, P.D. Shenoy, M.C. Mohan, K.R. Venugopal, Aspect term extraction for sentiment analysis in large movie reviews using Gini index feature selection method and SVM classifier. World Wide Web **20**, 135–154 (2017)
108. H. Markowitz, Portfolio selection. J. Financ. **7**, 77–91 (1952)
109. S. Marsella, J. Gratch, Computationally modeling human emotion. Commun. ACM **57**(12), 56–67 (2014)
110. J. Martineau, T. Finin, Delta TFIDF: an improved feature space for sentiment analysis, in *Proceedings of the Third International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2009, pp. 258–261
111. P. Melville, W. Gryc, R.D. Lawrence, Sentiment analysis of blogs by combining lexical knowledge with text classification, in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 1275–1284
112. R.C. Merton, On estimating the expected return on the market: an exploratory investigation. J. Financ. Econ. **8**(4), 323–361 (1980)
113. D. Metzler, W.B. Croft, A Markov random field model for term dependencies, in *Proceedings of the 28th Annual International Conference on Research and Development in Information Retrieval (SIGIR)*, 2005, pp. 472–479
114. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*, 2013, pp. 3111–3119
115. S. Minhas, A. Hussain, From spin to swindle: identifying falsification in financial text. Cogn. Comput. **8**(4), 729–745 (2016)
116. M. Minsky, *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind* (Simon & Schuster Paperbacks, Princeton, 2007)
117. S.M. Mohammad, S. Kiritchenko, X. Zhu, NRC-Canada: building the state-of-the-art in sentiment analysis of tweets, in *Proceedings of the Seventh International Workshop on Semantic Evaluation Exercises (SemEval)*, 2013

118. A. Moore, P. Rayson, S. Young, Domain adaptation using stock market prices to refine sentiment dictionaries, in *Proceedings of ESA, LREC Workshop*, 2016, pp. 63–66
119. S. Morris, A. Postlewaite, H.S. Shin, Depth of knowledge and the effect of higher order uncertainty. Econ. Theory **6**(3), 453–467 (1995)
120. A.K. Nassirtoussi, S. Aghabozorgi, T.Y. Waha, D.C.L. Ngo, Text mining for market prediction: a systematic review. Expert Syst. Appl. **41**, 7653–7670 (2014)
121. N.M. Neykov, P. Filzmoser, P.N. Neytchev, Robust joint modeling of mean and dispersion through trimming. Comput. Stat. Data Anal. **56**(1), 34–48 (2012)
122. T.H. Nguyen, K. Shirai, Topic modeling based sentiment analysis on social media for stock market prediction, in *The 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015, pp. 1354–1364
123. T.H. Nguyen, K. Shirai, J. Velcin, Sentiment analysis on social media for stock movement prediction. Expert Syst. Appl. **42**, 9603–9611 (2015)
124. M. Nofer, O. Hinz, Using Twitter to predict the stock market: where is the mood effect? Bus. Inf. Syst. Eng. **57**(4), 229–242 (2015)
125. P. Nystrup, B.W. Hansen, H. Madsen, E. Lindstrom, Regime based versus static asset allocation: letting the data speak. J. Portf. Manag. **42**(1), 103–109 (2015)
126. N. Ofek, S. Poria, L. Rokach, E. Cambria, A. Hussain, A. Shabtai, Unsupervised commonsense knowledge enrichment for domain-specific sentiment analysis. Cogn. Comput. **8**(3), 467–477 (2016)
127. J. Owyang, The future of the social web, 2009, Technical Report available at https://pacoprieto.files.wordpress.com/2009/09/futureofthesocialweb.pdf
128. A. Panagiotelis, C. Czado, H. Joe, J. Stoeber, Model selection for discrete regular vine copulas. Comput. Stat. Data Anal. **106**, 138–152 (2017)
129. B. Pang, L. Lee, Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005, pp. 115–124
130. P.N. Pant, W.H. Starbuck, Innocents in the forest: forecasting and research methods. J. Manag. **16**(2), 433–460 (1990)
131. S. Pinker, Clarifying the logical problem of language acquisition. J. Child Lang. **31**(4), 949–953 (2004)
132. S. Pinker, *How the Mind Works* (W. W. Norton & Company, New York, 2009)
133. R. Plutchik, The nature of emotions: human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. Am. Sci. **89**(4), 344–350 (2001)
134. S. Poria, E. Cambria, A. Gelbukh, F. Bisio, A. Hussain, Sentiment data flow analysis by means of dynamic linguistic patterns. IEEE Comput. Intell. Mag. **10**(4), 26–36 (2015)
135. S. Poria, E. Cambria, D. Hazarika, P. Vij, A deeper look into sarcastic tweets using deep convolutional neural networks, in *International Conference on Computational Linguistics (COLING)*, 2016, pp. 1601–1612
136. S. Poria, E. Cambria, G. Winterstein, G.-B. Huang, Sentic patterns: dependency-based rules for concept-level sentiment analysis. Knowl. Based Syst. **69**, 45–63 (2014)
137. B. Qian, K. Rasheed, Hurst exponent and financial market predictability, in *Proceedings of the 2nd International Conference on Financial Engineering and Applications*, 2004
138. H. Qiu, F. Han, H. Liu, B. Caffo, Robust portfolio optimization, in *Neural Information Processing Systems (NIPS)*, 2015, pp. 46–54
139. S.T. Rachev, S.V. Stoyanov, A. Biglova, F.J. Fabozzi, An empirical examination of daily stock return distributions for U.S. Stocks, in *Data Analysis and Decision Support* (Springer, Berlin/Heidelberg, 2005), pp. 269–281
140. G. Rachlin, M. Last, D. Alberg, A. Kandel, Admiral: a data mining based financial trading system, in *IEEE Symposium on Computational Intelligence and Data Mining*, 2007
141. Y. Ren, Y. Zhang, M. Zhang, D. Ji, Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings, in *Proceedings of the Thirty-th AAAI Conference on Artificial Intelligence (AAAI)*, 2016, pp. 3038–3044

142. S. Rothmann, E.P. Coetzer, The big five personality dimensions and job performance. SA J. Ind. Psychol. **29**(1), 68–74 (2003)
143. E.J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, A. Jaimes, Correlating financial time series with micro-blogging activity, in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, 2012, pp. 513–522
144. I.A. Sag, T. Baldwin, F. Bond, A. Copestake, D. Flickinger, Multiword expressions: a pain in the neck for NLP. Lect. Notes Comput. Sci **2276**, 1–15 (2002)
145. H. Sakaji, R. Murono, H. Sakai, J. Bennett, K. Izumi, Discovery of rare causal knowledge from financial statement summaries, in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–7
146. S. Satchell, A. Scowcroft, A demystification of the black-Litterman model: managing quantitative and traditional portfolio construction. J. Asset Manag. **1**(2), 138–150 (2000)
147. C. Schmitt, D. Dengler, M. Bauer, Multivariate preference models and decision making with the MAUT machine, in *International Conference on User Modeling (UM)*, 2003, pp. 297–302
148. R.P. Schumaker, H. Chen, Textual analysis of stock market prediction using breaking financial news: the AZFinText system. ACM Trans. Inf. Syst. **27**(2), 12:1–12:19 (2009)
149. R.P. Schumaker, Y. Zhang, C.-N. Huang, H. Chen, Financial fraud detection using vocal, linguistic and financial cues. Decis. Support. Syst. **53**, 458–464 (2012)
150. K. Schweser, *Schwesernotes 2019 Level 1 CFA Book 4: Corporate Finance, Portfolio Management, Equity Investments* (Kaplan Inc, La Crosse, 2018)
151. S. Shacham, A shortened version of the profile of mood states. J. Pers. Assess. **47**(3), 305–306 (1983)
152. S. Shalev-Shwartz, N. Srebro, SVM optimization: inverse dependence on training set size, in *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML)*, 2008, pp. 928–935
153. W. Shen, J. Wang, Portfolio selection via subset resampling, in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, San Francisco, 2017, pp. 1517–1523
154. J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, X. Deng, Exploiting topic based Twitter sentiment for stock prediction, in *The 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sofia, 2013, pp. 24–29
155. J. Si, A. Mukherjee, B. Liu, S.J. Pan, Q. Li, H. Li, Exploiting social relations and sentiment for stock prediction, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1139–1145
156. P. Sironi, *FinTech Innovation: From Robo-Advisors to Goal Based Investing and Gamification* (Wiley, Chichester, 2016)
157. J. Smailović, *Sentiment Analysis in Streams of Microblogging Posts*. Ph.D thesis, Jožef Stefan Institute, 2014
158. F.A. Sortino, L.N. Price, Performance measurement in a downside risk framework. J. Invest. **3**, 59–64 (1994)
159. J.F. Sowa, Semantic networks, in *Encyclopedia of Artificial Intelligence* (Wiley, New York, 1987)
160. R. Speer, C. Havasi, Representing general relational knowledge in conceptnet 5, in *Language Resources and Evaluation Conference (LREC)*, 2012, pp. 3679–3686
161. M. Spies, An ontology modelling perspective on business reporting. Inf. Syst. **35**, 404–416 (2010)
162. C. Strapparava, A. Valitutti, Wordnet-affect: an affective extension of wordnet, in *Language Resources and Evaluation Conference (LREC)*, 2004
163. I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in *Advances in Neural Information Processing Systems (NIPS)* (Trans Tech Publications, Switzerland, 2014), pp. 3104–3112
164. M. Taboada, J. Brooke, M. Tofiloski, K.D. Voll, M. Stede, Lexicon-based methods for sentiment analysis. Comput. Linguist. **37**(2), 267–307 (2011)
165. Y. Tai, H. Kao, Automatic domain-specific sentiment lexicon generation with label propagation, in *The 15th International Conference on Information Integration and Web-Based Applications & Services*, 2013, p. 53

166. N.N. Taleb, Finiteness of variance is irrelevant in the practice of quantitative finance. Complexity **14**(3), 66–76 (2008)
167. D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin, Learning sentiment-specific word embedding for Twitter sentiment classification, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014, pp. 1555–1565
168. P.C. Tetlock, M. Saar-Tsechansky, S. Macskassy, More than words: quantifying language to measure firms' fundamentals. J. Financ. **63**(3), 1437–1467 (2008)
169. T. Tieleman, G.E. Hinton, Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning (2012)
170. D. Tran, D.M. Blei, E.M. Airoldi, Copula variational inference, in *Advances in Neural Information Processing Systems (NIPS)* (Springer, Cham, 2015), pp. 3564–3572
171. R.R. Trippi, J.K. Lee, *Artificial Intelligence in Finance & Investing* (Irwin Professional Publishing, Chicago, 1996)
172. R.S. Tsay, *Analysis of Financial Time Series*, 2 edn. (Wiley-Interscience, Hoboken, 2005)
173. A.M. Turing, Computing machinery and intelligence. Mind **LIX**(236), 433–460 (1950)
174. P.D. Turney, Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews, in *The 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 417–424
175. M. Uhl, Reuters sentiment and stock returns. J. Behav. Financ. **15**(4), 287–298 (2014)
176. U.S. Bank, Our approach to asset allocation. Private Wealth Management Insights (2014)
177. R. Valitutti, Wordnet-affect: an affective extension of wordnet, in *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, 2004, pp. 1083–1086
178. I.S. Vicente, R. Agerri, G. Rigau, Simple, robust and (almost) unsupervised generation of polarity lexicons for multiple languages, in *European Chapter of the Association for Computational Linguistics (EACL)*, 2014, pp. 88–97
179. D.-T. Vo, Y. Zhang, Don't count, predict! an automatic approach to learning sentiment lexicons for short text, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 219–224
180. Y. Wang, Y. Zhang, B. Liu, Sentiment lexicon expansion based on neural pu learning, double dictionary lookup, and polarity association, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 553–563
181. W. Wei, Y. Mao, B. Wang, Twitter volume spikes and stock options pricing. Comput. Commun. **73**, 271–281 (2016)
182. A. Weichselbraun, S. Gindl, F. Fischer, S. Vakulenko, A. Scharl, Aspect-based extraction and analysis of affective knowledge from social media streams. IEEE Intell. Syst. **32**(3), 80–88 (2017)
183. R.E. Welsch, X. Zhou, Application of robust statistics to asset allocation models. Revstat Stat. J. **5**(1), 97–114 (2007)
184. T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, S. Patwardhan, Opinionfinder: a system for subjectivity analysis, in *Empirical Methods in Natural Language Processing (EMNLP)* (Springer, New York, 2005)
185. F. Wu, Y. Huang, Sentiment domain adaptation with multiple sources, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 301–310
186. F. Wu, Y. Huang, J. Yan, Active sentiment domain adaptation, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 1701–1711
187. B. Wuthrich, V. Cho, S. Leung, D. Permunetilleke, K. Sankaran, J. Zhang, Daily stock market forecast from textual web data, in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, 1998, pp. 2720–2725
188. F.Z. Xing, E. Cambria, L. Malandri, C. Vercellis, Discovering Bayesian market views for intelligent asset allocation, in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2018
189. F.Z. Xing, E. Cambria, R.E. Welsch, Intelligent Bayesian asset allocation via market sentiment views. IEEE Comput. Intell. Mag. **13**(4), 25–34 (2018)

190. F.Z. Xing, E. Cambria, R.E. Welsch, Natural language based financial forecasting: a survey. Artif. Intell. Rev. **50**(1), 49–73 (2018)
191. F.Z. Xing, E. Cambria, R.E. Welsch, Growing semantic vines for robust asset allocation. Knowl. Based Syst. **165**, 297–305 (2019)
192. F.Z. Xing, E. Cambria, Y. Zhang, Sentiment-aware volatility forecasting. Knowl. Based Syst. **176**, 68–76 (2019)
193. F.Z. Xing, E. Cambria, X. Zou, Predicting evolving chaotic time series with fuzzy neural networks, in *International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3176–3183
194. F.Z. Xing, F. Pallucchini, E. Cambria, Cognitive-inspired domain adaptation of sentiment lexicons. Inf. Process. Manag. **56**, 554–564 (2019)
195. F.Z. Xing, Y. Xu, A logistic regression model of irony detection in Chinese internet texts. Res. Comput. Sci. **90**, 239–249 (2015)
196. J. Xue, E. Zhu, Q. Liu, J. Yin, Group recommendation based on financial social network for robo-advisor. IEEE Access **6**, 54527–54535 (2018)
197. H. Yoon, H. Zo, A.P. Ciganek, Does XBRL adoption reduce information asymmetry? J. Bus. Res. **64**, 157–163 (2011)
198. A. Yoshihara, K. Seki, K. Uehara, Leveraging temporal properties of news events for stock market prediction. Artif. Intell. Res. **5**(1), 103–110 (2016)
199. G.P. Zhang, Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing **50**, 159–175 (2003)
200. L. Zhang, C. Aggarwal, G.-J. Qi, Stock price prediction via discovering multi-frequency trading patterns, in *The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 2141–2149
201. W. Zhang, S. Skiena, Trading strategies to exploit blog and news sentiment, in *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media (ICWSM)*, Washington, DC, 2010, pp. 375–378
202. X. Zhong, E. Cambria, Time expression recognition using a constituent-based tagging scheme, in *Proceedings of the 2018 World Wide Web Conference (WWW)*, 2018, pp. 983–992
203. X. Zhu, H. Guo, S. Mohammad, S. Kiritchenko, An empirical study on the effect of negation words on sentiment, in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014, pp. 304–313
204. Z. Zhu, R.E. Welsch, Robust dependence modeling for high-dimensional covariance matrices with financial applications. Ann. Appl. Stat. **12**(2), 1228–1249 (2018)

Index

F. Xing et al., *Intelligent Asset Management*, Socio-Affective Computing 9,
https://doi.org/10.1007/978-3-030-30263-4