

[9]/10

2014 - PhD thesis - 2021.10.14
250 pages 2021.10.31
[To BE completed] ... 2021.11.03

FREDERIK HOGENBOOM

Automated Detection of Financial Events in News Text

1.0	341,79	8	42
521,10			
779,00			
889,76			
670,90	6,0		
810,59	5,57	4,01	
881,00	3,06	2,59	
9.01,20	18,24	1,21	
7,150	9,32	1,10	
dey4,79	4,40	0,80	
1100,99	20,08	2,02	
126,76*	0,16	1,55	
516,12	X	1,32	
203,01	X	3,79	
US,02	0,17	2,09	
4410,00	45,62	1,20	
339,62	X	1,27	

2e nummer

in hier is de ols
te hoi en mee
en dit tyd
ter wene de
wige die de a
le heel wjecat
sociale com
de een ghet
sel.

de piej in
mida de te
grafische is
in alle kro
alle

618L MECH)
UVEZH IS
ML. AS?!



ALLEG!

een niek teai mitteken
M.M. van kann een van
in a. Nihau.
Klaar in momen in viss
molen uitkom kommeke en
momma 1970 nul in iguak
maka?

Hier sli in op dat er een
lange meer en steigerde
so met so lage en alles
van lama, dat hebje magie
alles niet want in maki's
mais

Wij zijn indische

→ a very comprehensive comparison of adv. vs disadv. of
data-driven vs. knowledge-based vs hybrid methods
(Statistical ML) (knowledge-based) (Syntax rules)

AUTOMATED DETECTION OF FINANCIAL EVENTS IN NEWS TEXT

Automated Detection of Financial Events in News Text

Automatische detectie van financiële gebeurtenissen in nieuwsberichten

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
rector magnificus

Prof.dr. H.A.P. Pols

and in accordance with the decision of the Doctorate Board

The public defense shall be held on

Thursday 11 December 2014 at 13:30 hours

by

FREDERIK PIETER HOGENBOOM
born in Dordrecht, The Netherlands.



Doctoral Committee

Promotors:	Prof.dr.ir. U. Kaymak Prof.dr. F.M.G. de Jong
Other members:	Prof.dr.ir. R. Dekker Prof.dr. P. Cimiano Prof.dr. A.P. de Vries
Copromotor:	Dr.ir. F. Frasincar

Erasmus Research Institute of Management - ERIM

The joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam
Internet: <http://www.erim.eur.nl>

ERIM Electronic Series Portal: <http://hdl.handle.net/1765/1>

ERIM PhD Series in Research in Management, 326

ERIM reference number: EPS-2014-326-LIS

ISBN: 978-90-5892-386-8

©2014, Frederik Hogenboom

Design: B&T Ontwerp en advies www.b-en-t.nl

Cover: Rebecca van den Heuvel

This publication (cover and interior) is printed by haveka.nl on recycled paper, Revive®.

The ink used is produced from renewable resources and alcohol free fountain solution.

Certifications for the paper and the printing production process: Recycle, EU Flower, FSC, ISO14001.

More info: <http://www.haveka.nl/greening>

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the author.





SIKS Dissertation Series No. 2014-41

The research reported in this thesis has been carried out in cooperation with
SIKS, the Dutch Research School for Information and Knowledge Systems
(<http://www.siks.nl>)

Preface

“The day when the scientist, no matter how devoted, may make significant progress alone and without material help, is past.”

Ernest Lawrence (1901 – 1958)

The road to the completion of a dissertation is not one to be walked alone. I am grateful to all people that helped me throughout my prolonged journey. First and foremost, I thank dr. Flavius Frasincar, who is not only a tremendously knowledgeable, enthusiastic, motivating, stimulating, and friendly daily supervisor, but also an awesome travel buddy. Without you, Flavius, conferences wouldn't have been half as fun! I am also greatly indebted to you for the time you have invested in organizing meetings, reading papers, and keeping track of my Ph.D. trajectory. I am not sure whether the ever incorrect time displayed on your watch played a significant and perhaps even underestimated role, but somehow you managed to squeeze more hours out of a day than most people can, and taught me how to focus, set goals, and be as productive as you are.

Furthermore, I would like to thank my promotores Prof.dr.ir. Uzay Kaymak and Prof.dr. Franciska de Jong. I really enjoyed our discussions, not only on the topic of my dissertation, but especially on the wide variety of subjects, ranging from fuzzy computing to e-humanities. These discussions often left me inspired and sparked new ideas. Many of them are still on my (ever-growing) to-do list! Uzay, I am also grateful that, although you found a new challenge in Eindhoven, your supervision continued as normal. Our meetings were always fruitful, and you often steered me in the right direction when I was lost in the academic wilderness. Franciska, thank you for your thorough reviews, which have always been valuable for improving my drafts. Your unique perspective helped me to view my work from new angles and levels of abstraction, and to see the large potential and wide applicability of my results. This opened doors to many interesting venues and meetings, which have been important for my academic development.

I am also indebted to the other members of my doctoral committee, i.e., Prof.dr.ir. Rommert Dekker, Prof.dr. Philipp Cimiano, and Prof.dr. Arjen de Vries for their valuable

input that helped me improving and perfecting this dissertation. Moreover, I would like to thank Prof.dr. Wolfgang Ketter for taking part in the large defense committee and for all past and ongoing efforts in our trading agent research. Last, I am thankful to have Prof.dr. Witold Abramowicz in my large defense committee at such a short notice.

Next, I would like to express my gratitude toward the Netherlands Organisation for Scientific Research (NWO) and the Dutch national program COMMIT, where my work was linked to the Physical Sciences Free Competition project 612.001.009: Financial Events Recognition in News for Algorithmic Trading (FERNAT), and the Infiniti project, respectively. Their substantial funding allowed me to conduct research and to visit conferences. Moreover, I would like to thank the Erasmus Research Institute of Management (ERIM) for their guidance throughout my trajectory, and for providing a good academic and social environment. Additionally, I thank the Dutch Research School for Information and Knowledge Systems (SIKS) for their support and amazing lectures.

Also, I am grateful for the daily help and support of the Econometric Institute supporting staff, without whom many things would not have run smoothly. Carien and Ursula, you have been marvelous office managers. Marjon and Marianne, thank you for diligently helping me out with my frequent, yet uncommon requests, and Anneke, I truly admire your patience and keen eye when it comes to my declarations. Antónia, I really enjoyed our conversations on Cabo Verde and life. Thank you for your good care at the tenth floor!

My colleagues from ERIM and Tinbergen have made my time at the Erasmus University Rotterdam a pleasant one. Particularly, I thank my brother and office mate Alexander, and also my friend and fellow Ph.D. candidate Damir for all the fun, lengthy conversations, support, sparring sessions, and fruitful collaborations that made Ph.D. life so much more enjoyable, not to mention the ever hilarious conferences (as Flavius' *trei purceluși*). Also, I am thankful that you guys took up the role of paranympths. Furthermore, I thank my (former) fellow Ph.D. candidates Kim, Charlie, Viorel, Tommi (who was technically never a Ph.D. candidate during my candidacy, but I have always considered him to be one of the guys), Rui, Nalan, Milan, Wim, Floris, and Joris for making the rough Ph.D. life bearable with entertaining breaks, interesting discussions, long drinks, etc. Also, I would like to thank the people from the Erasmus Ph.D. Association Rotterdam (EPAR), with whom I've worked several years, amongst which Margot, Geertjan, Marijn, Melek, Nufer, Jeanine, Jacqueline, Robert, Suzanne, Max, Alina, and Sergio. Thank you all for showing that there is much more to Ph.D. life than merely doing research!

A special thanks goes out to the (former) students of the Economics & Informatics Bachelor's and Master's programmes. In particular, I would like to mention Wouter,

Frank, Jordy, Michel, Marnix, Michael, Milan, Jeroen, Kevin, Arnout, Allard, Jethro, Wijnand, and Otto. Your hard and devoted work was a great source of inspiration, and our fruitful collaboration has resulted in many publications. Also, supervising many of you was a great experience.

Completing this dissertation would not have been possible without support from home. I thank my parents, family, and friends for supporting my decision to pursue a Ph.D. and their useful advice over the years. Also, I would like to express my sincere gratitude to all friends actively restocking the office fridge with plenty of drinks. Last, I thank my girlfriend Rebecca for being supportive throughout the years, providing feedback at unearthly times, skilfully listening to ideas and theories, accepting my absences caused by my travelling schedules, and understanding the importance and the inevitability of finishing work in the evenings or weekends. Thank you all for your support and genuine interest in my research, and for providing me with the more than welcome distractions!

Rotterdam, November 2014

Frederik Hogenboom

Table of Contents

Preface	vii
1 Introduction	1
1.1 Research Objectives	4
1.2 Contributions	8
1.3 Outline	9
2 Techniques and Applications of Event Extraction	11
2.1 Introduction	12
2.2 Techniques	16
2.2.1 Data-Driven Event Extraction	18
2.2.2 Knowledge-Driven Event Extraction	21
2.2.3 Hybrid Event Extraction	24
2.3 Applications	25
2.4 Research Issues	27
2.5 Development	29
2.6 Evaluation	31
2.7 Conclusions	32
3 A Semantics-Based Event Extraction Framework	35
3.1 Introduction	36
3.2 Related Work	38
3.2.1 SemNews	38
3.2.2 ANNIE	39
3.2.3 CAFETIERE	40
3.2.4 KIM	40
3.2.5 Discussion	41
3.3 Financial Event Detection based on Semantics	42

3.3.1	Domain Ontology	44
3.3.2	English Tokenizer	44
3.3.3	Ontology Gazetteer	45
3.3.4	Sentence Splitter	47
3.3.5	Part-Of-Speech Tagger	47
3.3.6	Morphological Analyzer	48
3.3.7	Word Group Look-Up	48
3.3.8	Word Sense Disambiguator	49
3.3.9	Event Phrase Gazetteer	52
3.3.10	Event Pattern Recognition	53
3.3.11	Ontology Instantiator	55
3.4	Evaluation	56
3.5	Conclusions	59
4	Event Extraction Patterns	61
4.1	Introduction	62
4.2	Related Work	65
4.2.1	Lexico-Syntactic Patterns	65
4.2.2	Lexico-Semantic Patterns	66
4.2.3	Pattern Learning	69
4.3	Hermes Information Extraction Language	71
4.3.1	Hermes	71
4.3.2	Language Syntax	72
4.3.3	Employing Ontology Elements	77
4.4	Rule Learning	78
4.4.1	Rule Learning Process	79
4.4.2	Representation	80
4.4.3	Initialization	80
4.4.4	Fitness Evaluation	81
4.4.5	Selection	81
4.4.6	Genetic Operations	81
4.4.7	Termination Criteria	83
4.5	Hermes Information Extraction Engine	83
4.5.1	Hermes News Portal	84
4.5.2	General Framework	84
4.5.3	Preprocessing	85

4.5.4	Rule Engine	87
4.5.5	Hermes Plug-in	89
4.6	Evaluation of Manually Created Patterns	90
4.6.1	Evaluation Setup	90
4.6.2	Evaluation Results	93
4.7	Evaluation of Automatically Learned Patterns	99
4.7.1	Evaluation Setup	99
4.7.2	Evaluation Results	101
4.8	Conclusions	103
4.A	Appendix: Hermes Information Extraction Language Grammar	105
5	Event-Driven Ontology Updating	109
5.1	Introduction	110
5.2	Related Work	111
5.3	OUL Syntax	112
5.3.1	Requesting Changes	113
5.3.2	Preconditions	113
5.3.3	Actions	114
5.3.4	Extensions	115
5.4	OUL Execution Models	116
5.4.1	Deferred and Immediate Updating	117
5.4.2	Matching First and All Changehandlers	119
5.4.3	Chaining Updates	122
5.4.4	Looping Updates	124
5.5	Implementation	126
5.6	Evaluation	127
5.7	Conclusions	128
6	Event-Based Stock Trading Strategies	131
6.1	Introduction	132
6.2	Related Work	134
6.3	News and Stock Markets	137
6.3.1	Event Information Extraction	138
6.3.2	Descriptive Statistics of the Data Set	138
6.3.3	News and Share Prices	139
6.4	Technical Trading	144
6.4.1	Simple Moving Average	145

6.4.2	Bollinger Bands	145
6.4.3	Exponential Moving Average	145
6.4.4	Rate of Change	146
6.4.5	Momentum	146
6.4.6	Moving Average Convergence Divergence	146
6.4.7	Performance of Technical Trading Indicators	146
6.5	A News-Based Trading Framework	147
6.6	Experiments and Results	150
6.6.1	Performance of Individual Events	150
6.6.2	News and Technical Indicators	151
6.6.3	Optimal Trading Strategies	152
6.7	Practical Considerations	156
6.8	Conclusions	157
7	Event-Based Risk Analysis	159
7.1	Introduction	160
7.2	Related Work	162
7.3	Event-Based Historical Value at Risk	164
7.4	Evaluation	169
7.4.1	Data	169
7.4.2	Metrics	169
7.4.3	Results	171
7.5	Conclusions	173
8	Conclusions and Outlook	175
8.1	Concluding Remarks	175
8.2	Outlook	177
Bibliography		181
Summary in English		207
Nederlandse Samenvatting (Summary in Dutch)		209
About the Author		211
ERIM Ph.D. Series Overview		213

List of Figures

Chapter 1

1.1	Apple's stock rates and major events in 2013	2
1.2	Topics of interest for event extraction	10

Chapter 2

2.1	A general architecture for a generic event extraction system	15
2.2	A qualitative evaluation of event extraction techniques	16
2.3	EVEX user interface for browsing large-scale databases for biomedical events	26
2.4	Hermes user interface for browsing news feeds for financial events	26

Chapter 3

3.1	SPEED design	43
3.2	A typical news example	43
3.3	<i>English Tokenizer</i> annotations (tokens)	44
3.4	Sample <i>OntoLookup</i> tree structure	46
3.5	<i>Ontology Gazetteer</i> annotations (concepts)	46
3.6	<i>Sentence Splitter</i> annotations (sentences)	47
3.7	<i>Word Group Look-Up</i> annotations (tokens)	49
3.8	<i>Event Phrase Gazetteer</i> annotations (phrases)	53
3.9	<i>Event Pattern Recognition</i> annotations (subject, predicate, and object) . .	55
3.10	<i>Event Pattern Recognition</i> annotations (events)	55

Chapter 4

4.1	Rule learning process	79
4.2	Overview of the Hermes processing pipeline	85
4.3	Rule tree template	88
4.4	Rule tree example	88

Chapter 5

5.1	A typical OULx changehandler for adding an item to an ontology	116
5.2	Deferred and immediate execution of a matching changehandler	119
5.3	Immediate execution of first and all matching changehandler(s)	121
5.4	Chained immediate execution of all matching changehandlers	124
5.5	Looped immediate execution of a matching changehandler	126

Chapter 6

6.1	Frequency of events in the data set	139
6.2	Example of a typical trading rule	148
6.3	News-based trading framework	150

Chapter 7

7.1	Overview of data flows and processing steps	165
7.2	Poisson distributions for measured and expected occurrences	167
7.3	Performance of event-based VaR prediction models	172

List of Tables

Chapter 3

3.1 Comparison of existing approaches and the characteristics required for our current endeavors	42
3.2 Common syntactic categories	48
3.3 Overview of the reported entity recognition precision and recall scores for several existing algorithms and information extraction pipelines	58

Chapter 4

4.1 Common lexical categories	74
4.2 Relations and events for the financial domain, used for evaluation purposes	91
4.3 Relations and events for the political domain, used for evaluation purposes	91
4.4 Creation times of lexico-syntactic and lexico-semantic rule groups in HIEL, and lexico-semantic rule groups in JAPE, using the financial test set	93
4.5 Creation times of lexico-syntactic and lexico-semantic rule groups in HIEL, and lexico-semantic rule groups in JAPE, using the political test set	93
4.6 Results of lexico-syntactic and lexico-semantic rule groups on the financial test set (within fixed time)	97
4.7 Results of lexico-syntactic and lexico-semantic rule groups on the political test set (within fixed time)	97
4.8 Inter-Annotator Agreement (IAA) for each of the considered relations	100
4.9 Results of HIEL rule groups after 5 hours of automatic rule learning and manual creation	101

Chapter 6

6.1 Average returns for different time intervals after an event	142
6.2 Abnormal returns for different time intervals after an event	143
6.3 Pearson's correlation between impact and returns, and between impact and abnormal returns for different time intervals after an event	144
6.4 Returns for signals generated by technical indicators	147

6.5	Returns for signals generated by news	151
6.6	Returns for signals generated by news and technical indicators	151
6.7	Optimal strategies for the FTSE350 data set	153
6.8	Optimal strategies for the S&P500 data set	155

Chapter 7

7.1	Examples of news event types identified by the ViewerPro software	166
7.2	Example VaR calculation for returns with and without noise cleaning	168
7.3	Comparison of the performance of traditional and event-based historical VaR calculations using a window of 8 hours	171
7.4	Comparison of the performance of traditional and event-based historical VaR calculation with rare events	172

List of Algorithms

Chapter 3

3.1 Word Sense Disambiguation	51
---	----

Chapter 5

5.1 Deferred ontology updating	118
5.2 Immediate ontology updating	118
5.3 Returning the first matching changehandler	120
5.4 Returning all matching changehandlers	120
5.5 Update collection from matched changehandlers	123
5.6 Update application from matched changehandlers	123
5.7 Looped deferred ontology updating	125
5.8 Looped immediate ontology updating	125

Chapter 6

6.1 Genetic programming approach for determining optimal trading strategies .	149
---	-----

Chapter 7

7.1 Stock price cleaning based on news events	167
---	-----

Chapter 1

Introduction

In modern day society, it is virtually impossible to maintain a successful business and to make proper decisions without being well-informed about all relevant market and societal developments. The tight coupling of business activities and numerical and textual data has been an incentive for the development of complex, interconnected, and multi-disciplinary systems, which have to deal with an exploding number of (digital) data sources and an ever-increasing stream of information and knowledge that can be extracted. Such systems typically are based on frameworks for automated data processing procedures which support decision making processes, and often also have a significant human-controlled component that ensures the system's integrity.

An important source of data is news, communicated by different media agencies through a variety of channels. Due to the abundance of news and the information contained within, it is becoming increasingly hard to timely and accurately deduce vital knowledge to support better informed decision making (Rampal, 1995). Given that news is time-sensitive, especially in the context of financial markets, selecting and processing all the relevant information in a decision-making process, is an extremely challenging task. An omnipresent problem relates to the weakly structured nature of news, which is presented using natural, human-understandable language, making the data limited in the degree to which it is machine-interpretable. This problem thwarts the automation of vital information and knowledge extraction processes – used for decision making – when involving large amounts of data.

Of utmost importance is the extraction of knowledge on financial *events*, which are phenomena captured in vocabulary pointing to specific (complex) concepts related to money and risk – like mergers and acquisitions, stock splits, dividend announcements, etc. – that can additionally be linked to actors, times, and places. Financial markets are extremely sensitive to breaking news (Chan, 2003; Ikenberry and Ramnath, 2002;

Michaely et al., 1995; Mitchell and Mulherin, 1994; Rosen, 2006). For instance, from January through August 2013, the stock rates of Apple Inc. (AAPL:NASDAQ) showed a difference of over 40% between the lowest (US\$390.53) and highest (US\$549.03) closing price, which can largely be attributed to the effects of Apple-related financial events that occurred during the same period. Figure 1.1 depicts stock rates (in US\$) of Apple Inc. over the course of 8 months in 2013. The stock rates are subject to rapid price swings, which are often paired with crucial financial events. Points A through G mark a few of these major financial events, and range from revenue announcements and acquisitions, to lawsuits and regulations.

In January 2013, Apple Inc. announced that the Q2 revenues were below expectations (point A), which immediately resulted in its shares to plummet by 12.4% overnight, when prices went down from US\$514.01 to US\$450.50. Two months later, the company was involved in a trial against Samsung Electronics, and lost (indicated by point B). This news immediately drove Apple's stocks down. After the stock rates showed the first signs of recovery, news on EU and Chinese regulators calling for the administration of iPad and iPhone distribution in Europe and a tighter supervision of Apple in China (points C and E, respectively), nullified the positive effects of news on Apple buying WiFiSlam (point D), only to recover over a month later when the California-based information technology

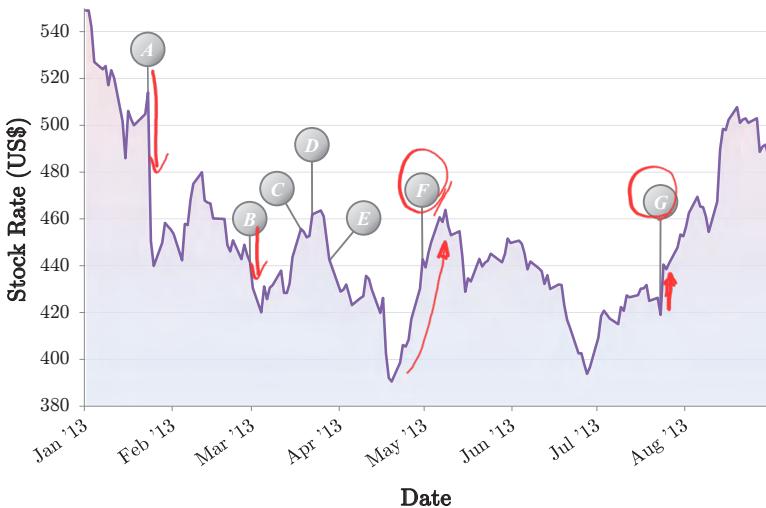


Figure 1.1: Stock rates (in US\$) and major events of Apple Inc. (AAPL:NASDAQ) for the observed period of January 2013 to August 2013.

company closed a US\$17 bln bond deal (point *F*). After a short drop in stock prices over summer, Apple quickly recovered when announcing the Q4 revenues, which met the expectations (point *G*), by jumping 5.1% overnight and 12.0% within a week.

This example shows that events contained within news messages can play an important role when exploited in financial applications, especially considering that, according to the weak form of the efficient market hypothesis, news containing information on an equity is not yet perfectly incorporated in the price when it is published (Fama, 1965). This can be taken advantage of in applications such as stock price movement prediction and algorithmic trading. Algorithmic trading represents the use of computer programs for entering trade orders with algorithms deciding on aspects like timing, price, and quantity of an order. Due to its low latencies (The Economist, 2007), already in 2007, algorithmic trading was estimated to represent 38% of U.S., 18% of European flow, and 4% of Asian flow (Berke, 2007), and these percentages have been increasing ever since. However, trading algorithms are currently mostly based on numerical inputs. Enhancing trading algorithms by considering financial events could thus yield improved profitability, and hence have a substantial impact on modern day trading.

Realizing the potential usefulness of news messages (and the events they describe) in financial applications, large news companies like Reuters, New York Times, and Dow Jones started to provide product services that offer tagged news items to be used for data-hungry tasks like algorithmic trading. Nowadays, such services are eagerly used by traders, who realise that keeping ahead of the competition by digesting and exploiting as much useful information as possible could yield serious profits. The current annotations provided by these vendors are *coarse-grained*, as they supply general information about the contents of news items, as for example company, topic, industry, etc., satisfying thus to a limited extent the information need in financial markets. For algorithmic trading, a *fine-grained* annotation (Drury and Almeida, 2011) that allows the identification of financial events as acquisitions, stock splits, dividend announcements, etc., is needed. Additionally, most annotations are merely based on article titles instead of full contents, and financial events (if any) are often not linked to semantic data structures, making reasoning and knowledge inference difficult. Therefore, we identify significant research opportunities to improve event extraction from news.

Automated event extraction, and especially semantically-enabled extraction, is intrinsically linked with well-known techniques from fields like Linguistics, Text Mining, Computer Science, Artificial Intelligence, and Knowledge Modeling. Common natural language processing and machine learning techniques are frequently deployed to solve crucial aspects of event extraction tasks, e.g., the recognition of named entities (such as

companies and persons), the disambiguation of word senses and references, etc. However, each of these techniques has its own challenges and difficulties, which accumulate through event extraction procedures, and which inevitably have their impact on the final solution quality.

Furthermore, although the application of event extraction in algorithmic trading is a promising direction, such automated trading activity itself has shown to result in risky practices. The infamous 2010 Flash Crash (Phillips, 2010), where trading algorithms caused a swing of almost 1,000 points in the Dow Jones Industrial Average stock market in a time span of several minutes, is just one of many examples of potentially dangerous consequences of (semi-)automated trading algorithms. Often, these trading algorithms execute trades without performing extensive safety checks due to the competitive time pressure, and their code is often not thoroughly tested before their release in real electronic markets, which could have serious consequences. Especially in light of recent developments, with the increasing popularity of news message processing and event extraction due to their many fruitful prospective applications, caution is well advised. Financial applications, such as algorithmic trading, could greatly benefit from the availability of a deeper insight into the reliability of automated event extraction procedures. In order to prevent other crashes such as the recent Twitter hoax spawned by hackers about an alleged bombing of the White House, erasing US\$200 bln of value from US stock markets in April 2013 (Lauricella et al., 2013), there is a need for further research in order to determine a principled way for accurate extraction of events from news.

1.1 Research Objectives

Because of the rich potential of event extraction from news for a wide variety of noteworthy financial applications, such as algorithmic trading and portfolio risk analysis, this dissertation addresses the problem of creating a semi-automatic and accurate financial event extraction framework for news messages. Hence, the problem statement underlying this dissertation is:

How to semi-automatically and accurately identify financial events in news messages, and how to effectively use such extracted events in financial applications?

We employ an interdisciplinary approach, where we combine well-known and established techniques from the fields of Linguistics, Text Mining, and Computational Intel-

ligence, but also rely heavily on state-of-the-art Semantic Web technologies to advance common applications in the field of Finance. The focus of this dissertation is primarily on knowledge-driven event extraction. The knowledge base (ontology) that forms the core of our methods has been developed, and is also (partially) maintained, by domain experts. Furthermore, the emphasis is on a semi-automatic approach to event extraction mainly geared toward the financial domain, although for generalizability purposes, other domains such as politics are additionally investigated.

In our endeavors, time-efficient and accurate event identification is achieved through the development of a robust, fast text processing pipeline based on superior components with proven high performance in terms of output quality on a wide variety of domains and data sets. Accuracy is maximized by considering the full body of news messages and making use of domain knowledge. Also, we improve the accuracy of financial event identification through the development and integration of an expressive event extraction pattern language that exploits domain knowledge. Moreover, we investigate the use of machine learning techniques to automatically learn patterns so as to lessen the amount of time and effort required for domain experts involved. Event extraction is supported by a knowledge base, which is updated automatically with newly extracted domain knowledge. This ensures that the knowledge base reflects the latest knowledge of the domain at hand, contributing to the overall accuracy of future event identification processes. Last, the usability of extracted events in financial applications is assessed in the context of systems for algorithmic trading and risk analysis, where we measure the effects of considering events based on performance improvements, e.g., higher profits or better risk estimations.

This dissertation answers a set of research questions that are related to techniques and applications for financial event extraction from news. The research questions resulting from the problem statement are explained next, along with an outline of their relevance and used methodology.

Question 1: What is the state-of-the-art for methods and systems for event extraction from text?

To date, there has been little overview work focusing on the field of event extraction from text. Therefore, in order to answer this research question, we review the current body of literature for techniques and applications for event extraction. First, we give an overview of the most popular high-performance event extraction techniques for textual data (distinguishing between data-driven, knowledge-driven, and hybrid methods) and present a multi-dimensional, qualitative evaluation of these. Next, we discuss applications

of event extraction from text corpora. Additionally, current research issues are identified and we provide some useful pointers to existing tools and libraries. Last, we discuss the evaluation methodology for event extraction systems.

Question 2: What is a suitable design for an automated, knowledge-driven event extraction system?

In order to extract financial events from news in a fast and accurate manner, one needs to construct a system (pipeline) for natural language processing, focusing on the accurate extraction and annotation of financial events at a speed that still enables real-time use. Therefore, in order to investigate this research question, we identify the key components of such a system, and develop a working implementation. Some of the components are existing, state-of-the-art parsers and processors made for specific tasks, such as tokenization and part-of-speech tagging. Additionally, most knowledge-driven components are newly created, as the number of such (freely available) components that are sufficiently fast and accurate is scarce. Quantitative evaluation of processing speed and accuracy is performed on the individual components, as well as on the pipeline as a whole.

Question 3: How can domain knowledge be used effectively for the extraction of (financial) events?

Within the context of the Semantic Web, domain knowledge is commonly stored in ontologies, describing concepts and their relations. Such ontologies can be of specific use in extraction patterns, which are often used in knowledge-based extraction frameworks. Concepts and their associated lexical representations are specified in one, central, place, thus simplifying pattern definitions by fostering the reuse of predefined concepts. Additionally, ontologies enable inference (reasoning) on concepts and relations, providing an additional layer of abstraction that can be exploited in extraction patterns.

Therefore, we define a lexico-semantic pattern language that, in addition to the lexical and syntactic information present in commonly used lexico-syntactic rules, also makes use of semantic information. Moreover, we investigate an evolutionary approach to pattern learning, which randomly and iteratively constructs and transforms sets of extraction patterns that are optimized based on their performances. After embedding the language and its learning algorithm into a news processing pipeline, the language is evaluated quantitatively on development times and result quality using a financial and a contrastive political data set, and is additionally compared to competitive lexico-syntactic and lexico-semantic alternatives. Furthermore, our rule learning approach is compared to the manual approach of pattern creation.

Question 4: How can extracted events be used for updating knowledge bases?

In knowledge-driven event extraction systems, it is vital to maintain an up-to-date knowledge base. Manual updating is tedious and time-consuming, and can thus become a bottleneck. Hence, we investigate an automated, event-driven approach to ontology updating so as to obtain a higher level of automation in event extraction systems. We extend an existing update language by using a trigger-based mechanism. Moreover, we propose different execution models, providing flexibility with respect to the update process. As a proof-of-concept, we implement the language and its execution models in an ontology-based news personalization service. We perform a qualitative evaluation of the various execution models in order to determine the best scenarios for each of the proposed models.

Question 5: How to utilize extracted events in algorithmic trading and financial risk analysis?

The extraction of events can be most useful if subsequent actions are undertaken that take into consideration the newly acquired knowledge. Therefore, in order to answer the research question at hand, we look into two highly relevant, practical financial applications, and determine whether the addition of events in their computations yields a substantial improvement in performance.

The first researched application of extracted financial events is algorithmic trading. After a study based on expert information, we determine the effects (impacts) of various financial events on future prices and revenues. Next, we identify popular numerical technical trading indicators for algorithmic trading and devise a way of converting events into numerical signals. Subsequently, we construct trading rules using an evolutionary algorithm, based on event signals and common technical trading indicators, by iteratively generating, mutating, duplicating, and mixing sets of random trading rules, while optimizing their performance. Evaluation is performed on two large financial data sets, and is based on expected revenues for various time horizons.

In our second application, we investigate the estimation of Value at Risk (VaR), a widely used method to assess portfolio risk. In order to improve the accuracy of VaR predictions, we consider extracted, rare events to be a possible cause of trend disruptions, due to the sensitivity of stocks to emerging news. First, we investigate how to distinguish rare events from regular events. Subsequently, we clean our data set from the disturbances generated by atypical rare events, and compare the accuracies of the proposed, event-corrected computations against those of the traditional VaR calculations in terms of mean squared errors, number of outperformances, and overconfidence. Additionally, we optimize the time window in which the data is cleaned through a systematic analysis of the employed performance indicators.

1.2 Contributions

The contribution of this dissertation is five-fold. For general information extraction, many surveys exist on popular, top-of-the-line techniques and their applications (Cowie and Lehnert, 1996; Hogenboom et al., 2009, 2010b). However, for event extraction, this is not the case (Hogenboom et al., 2011b, 2014c). Hence, our first contribution lies in providing an overview of the most popular high-performance event extraction techniques for textual data through a multi-dimensional, qualitative evaluation. Additionally, we identify common applications and research issues, discuss useful tools and libraries, and research techniques for the evaluation of event extraction systems.

Despite the impressive number and wide variety of available information extraction frameworks, only a handful of these systems focus on event extraction, or more specifically, focus on the extraction of financial events. Therefore, our second contribution is an ontology-based, knowledge-driven pipeline for extracting financial events (Hogenboom et al., 2011a, 2013b; Hogenboom, 2012; Hogenboom et al., 2010d, 2012d). The competitiveness of the pipeline is evaluated in terms of precision, recall, and F_1 scores, but also in execution times. Furthermore, the pipeline is flexible and its components are also reusable for other purposes in the broad spectrum of information extraction applications, such as web service discovery (Sangers et al., 2012a, 2013) and news recommendation (Capelle et al., 2012, 2013; Frasincar et al., 2009, 2011b; Goossen et al., 2011; Hogenboom et al., 2011c, 2014a,b; IJntema et al., 2010; Moerland et al., 2013; Schouten et al., 2010).

In many event-based applications, domain knowledge is readily available, but it is often not fully exploited. Therefore, our third contribution is a pattern language for extracting events (Hogenboom et al., 2012e; IJntema et al., 2012), with which domain experts can compose rules that make use of lexical, syntactic, and semantic elements. Not only do we present and evaluate a lexico-semantic pattern language that contributes to the state-of-the-art, but also we research and evaluate an evolutionary algorithm for learning patterns specified in this language (Hogenboom et al., 2013d; IJntema et al., 2014). Domain knowledge is fully exploited through ontological reasoning on semantic elements that are found in texts. The pattern language is not a graphical language, such as presented in our earlier published work (Hogenboom et al., 2010e,f, 2014d; Verheij et al., 2012b,c), nor are the patterns constructed visually through the selection of concepts (Frasincar et al., 2011a; Schouten et al., 2010), but it is an expressive, text-based language, similar to its early predecessors (Borsje et al., 2010; Hogenboom et al., 2013b). Although many pattern-based extraction languages are already available (Black et al., 2005; Hearst, 1992, 1998; Hung et al., 2010; Jacobs et al., 1991; Maynard et al., 2002; Saggion et al., 2007; Soderland,

1999), such languages often have a limited expressiveness, lack reasoning support due to the absence of formal semantics, are cumbersome in use, or are only suitable for extracting single entities but not (complex) events, which underlines the need for an expressive, knowledge-driven pattern language specifically aimed at event extraction.

For event-driven applications, an up-to-date knowledge base (ontology) is crucial to maximize accuracy. As manually updating is a time-consuming process, automated approaches to ontology updating are receiving increased attention. Currently, however, there is a lack of ontology update languages supporting a diverse set of execution models that enable for instance deferred, looped, or chained execution of updates. Hence, our fourth contribution consists of an automated, event-driven approach to ontology updating (Hogenboom et al., 2012f; Sangers et al., 2012b), which extends an existing ontology update language (Lösch et al., 2009) by using a trigger-based mechanism. Our proposed, implemented, and evaluated execution models provide flexibility with respect to the update process. Although our contributions do not take into consideration any execution order optimization (Hogenboom et al., 2012a, 2013a), we do provide the constructs that enable the composition of complex, event-driven ontology update mechanisms.

Since the 2010 Flash Crash, many recommendations have been made to prevent repetition of the disastrous events caused by fully-automated (trading) applications, e.g., a kill switch stopping execution at one or more levels, or various external limits and restrictions (Clark, 2012). However, the question remains whether these recommendations are sufficient. Therefore, with our event extraction framework, pattern language, and update language, we focus on improving the inputs – and herewith also the outputs – of common event-based applications, which can not only be deployed in fully automated applications such as high-frequency trading, but also in semi-automatic approaches that require human judgments in specific processing stages. To research the feasibility of considering events as additional inputs in financial applications, as a fifth and last contribution, we present two applications of extracted events, i.e., the development of algorithmic trading rules (Nuij et al., 2014) and the assessment of Value at Risk (Hogenboom et al., 2012b,c, 2013c). In both cases, the events have been accurately determined and hence are of high quality. We investigate how to transform events to signals and evaluate whether the addition of events to the applications’ inputs improves the overall performances.

1.3 Outline

The subsequent chapters answer (parts of) a research question, and are based on journal articles and other publications. Figure 1.2 depicts the main topics addressed in this

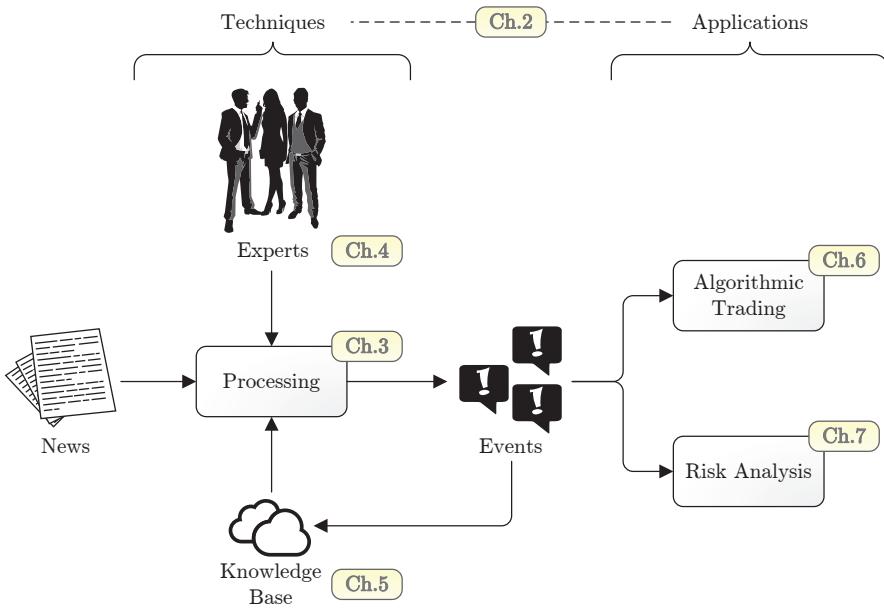


Figure 1.2: Topics of interest for event extraction.

dissertation, emphasising their inter-relatedness. The corresponding chapters are depicted as well. The rest of this dissertation, in which we mainly distinguish between event extraction techniques and event-based applications, is outlined as follows.

Chapter 2 presents an overview of the state-of-the-art techniques and applications of event extraction, and corresponds to our first research question. The subsequent three chapters discuss our proposed techniques related to event extraction. Chapter 3 provides an answer to our second research question, and discusses a framework for the knowledge-driven extraction of financial events. Chapter 4 investigates the third research question and proposes an expressive pattern language, incorporating expert knowledge, for event extraction, and additionally researches an evolutionary learning algorithm for automatic pattern construction. Chapter 5 is linked to the fourth research question, and proposes a way to employ events for knowledge base updating, hereby specifically focusing on trigger-based execution models. Chapters 6 and 7 answer the last research question by elaborating on applications of event extraction, where events serve as an (additional) input of trading algorithms and Value at Risk computations, respectively. Last, Chapter 8 concludes this dissertation and gives directions for future work.

Chapter 2

Techniques and Applications of Event Extraction*

EVENT extraction, a specialized stream of information extraction rooted back into the 1980s, has greatly gained in popularity in the last decade due to the advent of big data and the advances in the related fields of text mining and natural language processing. However, to date, there has been little overview work focusing on this particular field. Therefore, first and foremost, we give an overview of the most popular high-performance event extraction techniques for textual data (distinguishing between data-driven, knowledge-driven, and hybrid methods) and present a qualitative evaluation of these. Moreover, we discuss common applications of event extraction from text corpora. Additionally, current research issues are identified and we provide some useful pointers to existing tools and libraries. Last, we discuss the evaluation of event extraction systems.

*This chapter is based on the article “F. Hogenboom, F. Frasincar, U. Kaymak, and F. de Jong. A Survey of Event Extraction Methods from Text for Decision Support Systems. *Decision Support Systems*, 2014. Under Review.”

2.1 Introduction

Over the years, Information Extraction (IE) has become increasingly popular as a tool for a vast array of applications (Cowie and Lehnert, 1996). At first, the IE field was focused particularly on message understanding in newswires (Grishman and Sundheim, 1996). However, due to the onset of progressively larger digital data collections of various natural language text types such as news messages, articles, and web pages, researchers and practitioners require more and more advanced techniques, extract more information with greater accuracies and on a real-time basis, and operate on larger scales than ever before. Since the early 2000's, there has been a notable shift from general information extraction from digital collections – mainly extracting basic named entities like persons and organizations – toward more advanced forms of text mining, including Event Extraction (EE) that requires the handling of textual content or data describing complex relations between entities (Björne et al., 2010). This development has been fueled by the continuous advances in Text Mining (TM) and Natural Language Processing (NLP), the advent of big data, as well as the availability of (manually) annotated data sets that often serve as a basis for building extraction models.

Event extraction combines knowledge and experience from a number of domains, including computer science, linguistics, data mining, artificial intelligence, and knowledge modeling. It is commonly seen as the TM-aided extraction of complex combinations of relations between actors (entities), performed after executing a series of initial NLP steps. It is a form of IE, aimed at specific users, applications, and platforms, that results in more complex and detailed outputs than regular IE. Event extraction originates in the late 1980s, when the U.S. Defense Advanced Research Projects Agency (DARPA) boosted research into message understanding, aimed at automating the identification of terrorism-related events from newswires, a topic that has remained trending up until today.

With the exponential growth of digital collections, event extraction research has evolved greatly. Early mentions of modern event extraction can be found in the biomedical literature, where NLP techniques have been traditionally employed for discovering biological entities such as genes and proteins, but where the same techniques are now also widely used for identifying events involving these entities, e.g., gene expressions and protein bindings (Yakushiji et al., 2001). Gradually, event extraction has moved to other domains such as politics and finance, where events like parliament changes, elections, announcements, CEO changes, or acquisitions, are also comprised of sets of entities (e.g., persons, governments, countries, or companies) and their relations (e.g., leadership, competitor, ownership, etc.) (IJntema et al., 2012).

An event is roughly defined as something that happens or that is regarded as happening during a particular interval of time. Events can have multiple occurrences and are generally seen as incidents of substantial importance. In this work, we do not consider organized events such as soccer matches, scientific conferences, parties, etc., but we focus on mostly unexpected occurrences which preferably need to be acted upon. Such events are universally associated with state changes. However, per domain, their definition, complexity, and interpretation could greatly differ.

For instance, in the financial and political domains, events closely relate to (human-related) happenings described in news, e.g., '*Google acquires Motorola Mobility*' and '*French troops move into Central African Republic*', but often also include environmental events such as '*Chelyabinsk meteor injures 1,500 people in southern Ural region*' and '*Massive Thailand floods drive up hard disk prices*'. In the biomedical domain, events are commonly seen as newly discovered interactions between biological entities, which have been reported on in the literature, e.g., '*The narL gene product partially activates the nitrate reductase operon*' or '*PD98059 is a specific inhibitor of MAPK kinase 1*'. Such events are often more complex than those used in other domains, as they consider causes and consequences of relationships between events and their participants, and between macro and micro-events, which have far more subtle relationships than those found among regular, physical entities in most domains.

Irrespective of their domains, extracted events are associated with changes in the state of the current knowledge, and hence can be employed for decision making, prediction, or monitoring. The applications are numerous, such as generating trading signals for stock exchange markets, providing event-driven data integration in decision support systems, creating social media monitoring systems by police departments, etc. Hence, these developments render traders, managers, companies, and governments to be the users that immediately benefit from event extraction.

Despite the envisaged usefulness and wide prospective applicability of event extraction, several hurdles have to be overcome until event extraction is widely adopted as a supportive tool in practice. The main requirements that were trending in the nineties for information extraction (Cowie and Lehnert, 1996), are still applicable to event extraction today. For instance, the technologies should deliver sufficiently accurate results, as anything less than 90% precision would not stimulate adoption by industry. Furthermore, construction and processing costs should be minimized, and systems are preferred to be operable by non-specialists. These challenging requirements have led to many research efforts in the last decade, of which the main ideas are surveyed in this chapter.

By and large, most event extraction systems that have been reported on in the recent literature, are constructed using a similar, general architecture, which is depicted in Figure 2.1. A typical event extraction system makes use of text-based inputs, such as archival content, text feeds, or digital library contents. Raw data are subsequently structured into machine-interpretable chunks (tokens with associated features) by means of several NLP steps like tokenization, part-of-speech tagging, lemmatization, and word sense disambiguation. Next, additional TM procedures handle the preprocessed data and apply logics or heuristics in order to extract entities, relations, events, and their properties. In most modern systems, a knowledge base is employed in one or more NLP and TM steps. Sometimes, systems employ a feedback loop, in which newly derived knowledge is immediately incorporated into the active knowledge base. Often, the initial NLP steps are crucial for the performance of event extraction systems. If, for instance, parts-of-speech (e.g., verbs, nouns, etc.) have not been correctly assigned, or word senses have not been disambiguated properly, then errors are propagated into the subsequent steps, hence resulting in erroneous results and tampering the performance of these systems. Event extraction systems have various outputs. Some outputs, such as knowledge base update statements (e.g., adding newly introduced products), remain hidden to the user, and can also be used as inputs for other systems. Others are visible to the user, and can be visualized as ordinary lists of identified events (formatted using templates, e.g., n-tuples such as '*(Google, introduces, Nexus 10 tablet)*' or '*(Tim Cook, becomes new CEO of, Apple)*'), signals (e.g., '*buy*' or '*sell*'), or annotations in the original documents (marking the discovered locations of the identified events in the text).

Evaluation of event extraction systems is usually done quantitatively, based on the classic precision, recall, and F_1 scores (van Rijsbergen, 1979), measuring the proportion of retrieved events that is relevant, the fraction of relevant retrieved event instances, and their harmonic mean, respectively. However, comparing systems remains a non-trivial task, as evaluations are often performed based on different, non-comparable data sets. Alternatively, one could measure the performance of event extraction systems in a qualitative manner, for example by evaluating features such as the amount of required data, knowledge, and expertise on the one hand, and the interpretability of the results (i.e., the amount to which results can be explained and traced), the required development (training) time, and the required execution time on the other hand. Also, user feedback could be monitored and assessed.

While IE in general is certainly a heavily researched and well-described area, to our knowledge there is little overview work focusing on the upcoming field of event extraction. Therefore, we focus on event extraction, with specific attention to high-performance ex-

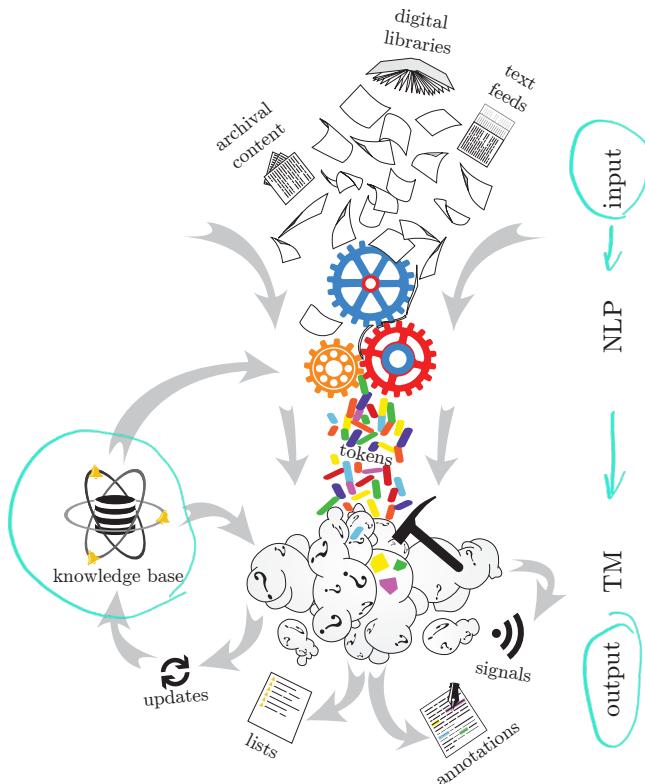


Figure 2.1: A general architecture for a generic event extraction system.

traction techniques and their common applications. Although for IE, the focus of recent work is gradually shifting toward open domain extraction (Etzioni et al., 2008), the main focus of this chapter is on domain-specific event extraction. From a commercial point of view, open-domain event extraction would be valuable, yet from a research perspective, it is difficult to evaluate the performance of such systems because of the need for annotated data. Therefore, most event extraction work focuses on domain-specific applications. Also, although a few recent event extraction efforts have expanded to non-textual sources such as transaction logs, video, or click behaviour, we focus on the core of event extraction, which only considers textual sources.

While a preliminary survey on event extraction from text already exists (Hogenboom et al., 2011b), here we provide a more complete overview on a higher level of abstraction, and also cover the most recent works. Moreover, in our current endeavors, the various approaches to event extraction are evaluated on more (qualitative) dimensions. We addi-

tionally discuss a typical system architecture, common applications, and current research issues to event extraction from text. Last, we provide some useful pointers to existing tools and libraries, and discuss the evaluation of event extraction systems.

2.2 Techniques

Both in recent research and in practice, a great many of different event extraction techniques have been described and applied. In the following discussion on the main techniques that are employed for event extraction, we omit the peculiarities of individual approaches, and focus on several aspects of various commonly applied extraction techniques, positioning the discerned event extraction methods in a multi-dimensional space and identifying their unique properties, advantages, and disadvantages.

In this overview, we deliberately refrain from comparing any of the discussed techniques based on reported quantitative measures, such as precision, recall, and F_1 scores, as most methods have been evaluated based on different data sets, arguably rendering a fair comparison unfeasible. Moreover, some experimental techniques have not yet been benchmarked on (standard) data sets, or have been evaluated using less common measures. Also, with enough fine-tuning, one could squeeze out competitive performances for most approaches. As there is no single-best approach to event extraction, we perform an alternative qualitative evaluation of event extraction techniques, by arranging the discussed example works in two multi-dimensional grids with ordinal scales, as depicted in Figure 2.2, highlighting the different aspects of each of the considered approaches.

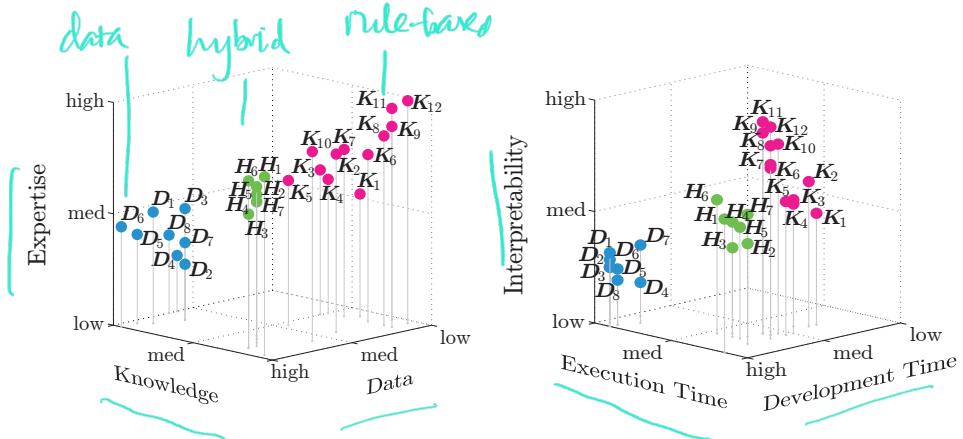


Figure 2.2: A qualitative evaluation of event extraction techniques.

Each grid explores three dimensions along the axes. The qualitative measures have been divided along the grids and axes in a way that ensures maximum separability. The first grid distinguishes between the amount of required data, knowledge, and expertise, whereas the second grid discerns the result interpretability, as well as the required development and execution time. In the figure, data points represent the examples discussed in our review and are labeled in order of appearance. The techniques are positioned along the axes based on insights gained from reported performances. Although on one dimension, differences are not always clear-cut, the multi-dimensionality of our analysis enables us to discern clearly distinct clusters, which overlap in some, but not all dimensions. The clusters have been assigned different colors and label prefixes, and represent the three main approaches to event extraction, discussed below.

The evaluated event extraction techniques are scaled relatively to one another on the six qualitative dimensions. The amount of required data is determined based on reported amounts of documents for which significant results are obtained, where low amounts total to no more than a couple of hundreds of documents, mid-range amounts represent around ten thousand documents, and, in case more documents are required, data points move towards the highest classification in the grid. Next, the amount of required knowledge is measured by evaluating the domain specificity of the methods, i.e., the amount of required domain knowledge for executing the necessary extraction steps. This amount is directly proportional to the number of steps requiring domain knowledge, and inversely proportional to the number of commonly applicable (universal) methods employed in a single event extraction approach. Hence, a higher number of universal methods lowers the score, while methods with an emphasis on domain knowledge receive higher scores. Subsequently, the amount of required expertise is determined by analyzing the number of methods that are combined. Also, the (computational) complexity of the methods themselves, the number of required steps, the length of the employed algorithms, etc., exert a notable influence on the amount of required expertise. Result interpretability is observed by evaluating the comprehensibility and traceability of the considered methods. The comprehensibility of the output, i.e., the ease with which results can be translated to human understandable language, is relatively low for numerical outputs, higher for lexical results, and maximal for outcomes that incorporate a strong notion of semantics. Black-box methods yield low traceability scores, whereas methods with results that can easily be backtracked (as is the case for grey-box, and – to a wider extent – white-box methods) yield much higher scores. Subsequently, the development time is composed of time invested in (model) construction, training, and parameter tweaking. Last, execution times are scaled based on reported run times or computational complexities.

A common distinction of event extraction approaches stems from the popular classification scheme in the field of modeling. Data-driven approaches, depicted as blue points labeled $D_{\#}$, aim to convert data to knowledge through the usage of statistics, data mining, and machine learning. Expert knowledge-driven methods (pink points, tagged with $K_{\#}$), extract knowledge by exploiting existing expert knowledge, usually through pattern-based approaches. In today's advanced extraction procedures, it is often the case that researchers do not explicitly opt for either a data-driven or a knowledge-driven approach in the NLP and TM stages of event extraction, but employ techniques from both fields by bootstrapping or optimizing their knowledge-based algorithms using machine learning, or vice-versa. However, most approaches are still mostly data-driven or knowledge-driven. Those extraction methods that equally employ data and knowledge-driven techniques can be categorized under the increasingly popular hybrid event extraction approaches, which have been marked in the figure with a green colour, labeled $H_{\#}$.

2.2.1 Data-Driven Event Extraction

The vast majority of event extraction tools make use of at least some data-driven techniques, and many of these tools even rely solely on quantitative methods to discover relations. Data-driven approaches require large text corpora in order to develop models that approximate linguistic phenomena. Such event extraction techniques are not restricted to basic statistical reasoning based on probability theory, but encompass all quantitative approaches to automated language processing, such as probabilistic modeling, information theory, and linear algebra. These methods focus on specific features, such as words and n -grams, as well as their associated weights, which are mostly determined using frequency counting algorithms. These features and their associated weights represent the input of complex clustering or classification algorithms, which, despite their differences, all focus on discovering statistical relations, i.e., facts that are supported by statistical evidence. These discovered relations, however, are not necessarily semantically valid, as semantics (meanings) are not explicitly considered, but are assumed to be implicit in the data.

For data-driven event extraction, there is a clear distinction between supervised and unsupervised learning approaches. The former approaches require some expert knowledge, as labeled data is provided to learning algorithms, whereas the latter approaches are usually employed when no labeled data is available. Unsupervised learning is commonly applied in data exploration or structure discovery tasks, and comprises techniques such as clustering and manifold learning. Supervised learning techniques typically produce new events, based on given labeled examples. Such algorithms deduce event properties

and characteristics from training data, and use these to generalize to unseen situations. Combining labeled and unlabeled data can produce considerable improvements in learning accuracy, and hence semi-supervised learning methods are often employed when there is a small amount of labeled data, and a large amount of unlabeled data available, for instance when dealing with special, expensive devices or methods (e.g., in bioinformatics).

Popular (supervised) machine learning techniques employed for learning relations, such as decision trees or neural networks, are often difficult to train for event extraction, because these methods require a large amount of data to be trained on, of which much is initially not labeled (annotated). Moreover, the number of negatives (irrelevant data points) tends to largely outweigh the number of positives (relevant data points) in these data sets, which does not only make the number of useful data points sparse, but also adds noise (or a bias) to the trained models. Many techniques exist for tackling this issue of unbalanced data, e.g., over-sampling (duplicating data), under-sampling (removing data), synthetic minority over-sampling (generating synthetic samples), etc. Usually, excessive amounts of negative examples are pruned first from the data, before training extraction models. For instance, in the literature, an event extraction method using support vector machines is reported, which supports the pruning of negative examples (Lei et al., 2005). This approach, denoted with D_1 in Figure 2.2, requires a vast amount of data in order to have enough positive examples for a reliable fit, yet filtering out the surplus of negatives improves interpretability of the results. The latter improvement, however, comes at the cost of an increased training time and required expertise, due to the additional filtering step. Moreover, the interpretability of the results by experts is still lower than for decision trees, which provide a much more natural representation of learned rules.

Another approach to data-driven event extraction is related to inference models, which are very popular in regular IE tasks. These models are mainly used in semi-supervised or unsupervised settings, usually operate on words in sentences or documents, and apply inference on a specific (learned) probability distribution. The latter distribution is used for predicting the next word in a sentence or document, based on the history of words. For instance, from a corpus it could show that ‘*ACM*’ is frequently followed by ‘*Press*’, but not so much by ‘*publishing*’, yielding a higher probability for ‘*Press*’ to follow ‘*ACM*’ in unseen texts. Inference models use the classification of previous words to predict the next word, by learning which words tend to follow specific words. A commonly used probabilistic model is the n -gram model, where the last word of the n -gram represents the word to be predicted. An example application can be found in (Miwa et al., 2010) (data point D_2). However, for low n values (i.e., small amounts of words are considered to be predictive for subsequent words), the amount of generated n -grams rapidly grows when

data sets get larger. Conversely, for high values of n , the risk of data sparseness increases. Hence, for event extraction applications, a method for learning a joint probabilistic model over events in sentences is proposed, which is in essence a statistical learning language based on First Order Logic and Markov logic (Riedel et al., 2009) (labeled D_3).

Clustering of similar or related documents, sentences, terms, etc., is a commonly employed, unsupervised data-driven technique for event extraction. There is no single-best clustering approach, as performance greatly depends on the scenario of use. Generally, one can choose between soft and hard clustering methods. Soft (or fuzzy) clustering methods allow for partial cluster memberships, which can be useful in linguistic tasks (such as assigning events to multiple event clusters, e.g., ‘*Square Enix announces increased profits and sales*’ to clusters of announcements, profit increases, and sales increases). Hard clustering only allows membership in one cluster (e.g., the latter event is an announcement, or an increase in profit, or an increase in sales), making it less useful for natural language processing. Examples of clustering-driven approaches are numerous. For instance, one could use clustering on event occurrences over time, and thus predict the type and properties of a new event (Okamoto and Kikuchi, 2009). Alternative options are clustering documents containing events (parsed through a shallow linguistic analysis) to identify events (Atkinson et al., 2009; Tanev et al., 2008), or sentences referring to the same event (Naughton et al., 2006). In more complex frameworks, clustering is usually combined with advanced graph structures, e.g., with weighted undirected bipartite graphs (Liu et al., 2008). The aforementioned clustering-based techniques have been incorporated into Figure 2.2, and are labeled from D_4 to D_8 , respectively.

As illustrated by Figure 2.2, the overall trend is that data-driven methods require a lot of data for their training in order to get statistically significant and reliable results. On the other hand, the role for expert knowledge is minimal, as these methods generally do not take into consideration domain semantics, but instead rely on universal, statistical methods that can be applied to any domain. In terms of expertise, however, there is not a straight line for data-driven event extraction, as the required expertise greatly depends on the methods that are applied. When combining multiple methods, the amount of required expertise is larger than when applying, for instance, merely a single, out-of-the-box clustering method. Also, (semi-)supervised methods generally require more expertise and expert knowledge, as labeled data is involved. Although training times are usually long, because of the excessive amounts of data that need to be processed on the one hand, and the computationally intensive operations involved on the other hand, execution times are mostly short for data-driven event extraction methods, as learned weights and parameters are applied to new examples without involving a lot of reasoning on a pre-

built model. Last, given the current limitations in the interpretability of machine learning results, we consider these methods to be opaque or semi-transparent at best. Moreover, the interpretability of the results of most data-driven methods is low, because results do not necessarily have explicit semantics associated.

2.2.2 Knowledge-Driven Event Extraction

In contrast to data-driven methods, knowledge-driven event extraction is often based on predefined (or sometimes learned) patterns that express rules representing expert knowledge. The TM procedures of knowledge-driven methods are hence inherently based on linguistic and lexicographic knowledge, as well as on existing human knowledge regarding the content of the texts to be processed. We can make a rough distinction between two types of patterns that can be applied to natural language corpora for event extraction, i.e., lexico-syntactic patterns (Aone and Ramos-Santacruz, 2000) and lexico-semantic patterns (Li et al., 2002). The former patterns are a combination of lexical representations and syntactic information. The latter patterns are more expressive, and combine lexical representations with both syntactic and semantic information.

Before extraction patterns are employed on a data set of natural language texts, in most knowledge-driven approaches, the corpus is preprocessed using data-driven or knowledge-driven parsers. Most patterns operate on so-called tokens, i.e., small text segments, which are usually words, word groups, numbers, spaces, or punctuation signs. These tokens get assigned various properties, depending on the level of detail and the focus of the natural language processing pipeline analyzing the corpus. Common properties are a token's associated semantic concept, lexical category (part-of-speech), orthographic category (capitalization), lemma (stem) with suffix and/or affix, pronominal reference, etc. Eventually, patterns, constructed according to a predefined grammar, are matched on large collections of tokens. Usually, in case of a match, additional data (properties) are collected and stored in data structures for later usage, e.g., subjects, objects, etc. In less sophisticated applications, only an event occurrence is registered.

When implementing knowledge-driven approaches to perform event extraction tasks, it is often difficult to stay within the boundaries of these approaches, and hence most methods often have a (small) data-driven component. For instance, initial clustering for classification could be required to determine usable elements for constructing patterns (e.g., proper nouns, verbs, companies, persons, etc.). In the following discussion on knowledge-driven approaches, we refer to approaches that are fully or mainly pattern-based, as this is the main characteristic of knowledge-driven event extraction methods.

Lexico-syntactic patterns often appear in earlier work on knowledge-driven event extraction, e.g., in (Aone and Ramos-Santacruz, 2000; Yakushiji et al., 2001) (points K_1 and K_2 in Figure 2.2), but have remained popular in more recent approaches (e.g., (Hung et al., 2010; Nishihara et al., 2009; Xu et al., 2006), labeled K_3 to K_5) due to their domain independency. The patterns mostly rely on syntactic properties (grammatical meanings) like verbs, nouns, prepositions, and pronouns. An example of a pattern that combines lexical representations and syntactic information is:

```

1 {NNP, }* NNP{,}? and NNP (announce | discuss)
2 collaboration {with NNP}?

```

This pattern can be used to mine a corpus for fusion and collaboration events of companies and/or persons, assuming that the proper nouns in the pattern (NNP) refer to companies or persons. More specifically, the pattern is used for finding occurrences in the text starting with zero or more comma-followed proper nouns, followed by a proper noun (with an optional additional comma), the text string and, and yet another proper noun. Next, nouns (i.e., companies or persons) that either announce or discuss a collaboration are required, and, optionally, this collaboration is followed by with and another proper noun.

This single pattern could cover a limited amount of different statements of negotiated or announced collaborations. Complex sentences with verb conjugations or sentences with different structures would not be matched due to the lack of synonyms and structure flexibility in the pattern. Clearly, defining the right event extraction patterns is not a trivial task. Ideally, patterns should be defined in such a way that they occur frequently and thus cover many event instances. The example rule could be improved by replacing (announce | discuss) with something more general, e.g., VB (base verb form). However, this also leads to erroneous matches that fall outside the scope of the intended event. Generalization hence comes at the cost of a loss in precision, although recall usually increases. As an alternative, one could enumerate all possible verbs and conjugations, but this greatly impacts development times and general rule flexibility. To cope with these and other related issues like synonymy, homonymy, and polysemy, very complex lexico-syntactic patterns need to be constructed. This stresses the need for higher-level patterns, such as lexico-semantic patterns.

Lexico-semantic patterns enable one to extract more accurate information from texts by enriching lexico-syntactic patterns through the addition of semantics, i.e., linguistic meaning and context. Lexico-semantic patterns allow for more powerful expressions without complicating the patterns too much in terms of number of elements, as these patterns leverage existing lexico-syntactic patterns to a higher abstraction level. In contrast to

lexico-syntactic patterns, however, some domain knowledge is required in order to create high-precision patterns that retrieve many events in an arbitrary corpus. This makes the creation of patterns less trivial, but on the other hand, because the patterns can be less general than is the case with lexico-syntactic patterns, they allow for a more specific description of one's needs and return more accurate results.

Following our running example, we can improve the pattern by making use of concepts, such as companies and persons. Hence, a lexico-semantic alternative of the lexico-syntactic pattern could be:

```

1 {([Company] | [Person]) _*}+ ([ToAnnounce] | [ToDiscuss])
2 [Collaboration] _* {[([Company] | [Person])}?

```

where the proper nouns have been replaced by **Company** and **Person** concepts (surrounded by square brackets), the verbs along with their variants and conjugations are stored in their respective **ToAnnounce** and **ToDiscuss** concepts, and the collaboration is captured by the **Collaboration** concept. For further simplification, the rule also contains wildcards (`_*`), matching any sequence of words or punctuation. Precision of such lexico-semantic patterns is generally higher than their lexico-syntactic counterparts, without losing much on the recall level.

In literature, there are two notions of lexico-semantic patterns. Some research is focused on event extraction by means of basic semantics, added through gazetteers (word lists) that are iteratively searched while parsing corpora. As moving from lexico-syntactic approaches to such simple-typed lexico-semantic approaches is a minor incremental step, this approach has often been used, and is for instance exemplified in (Atkinson et al., 2013; Capet et al., 2008) (points K_6 and K_7 in Figure 2.2).

Other research aims to use patterns based on ontological classes and relations, which capture the domain semantics. Ontology-based lexico-semantic patterns involve a more complex typing, as their elements capture domain semantics and are more advanced than syntactic and simple-typed semantic elements. Additionally, restrictions and relations applying to concepts specified in the underlying ontology can be utilized when applying reasoning with an inference engine. Compared to other pattern-based approaches, complex-typed patterns require more expertise due to the increased complexity, yet often generate better results due to their higher expressivity. Most complex-typed lexico-semantic languages, however, reduce their complexity by removing additional features that have been used frequently in lexico-syntactic languages, such as repetition operators or wildcards (exploited in our example), because they focus more on the usage of concepts. Examples of such works are numerous (Arendarenko and Kakkonen, 2012; Cohen et al.,

2009; Li et al., 2002; Vargas-Vera and Celjuska, 2004) and are depicted in Figure 2.2 as points K_8 to K_{11} . Some languages do offer full expressivity by not only considering ontological classes and relations, but also by additionally supporting labeling, negation, wildcards, and repetition operators (IJntema et al., 2012) (K_{12} in the figure).

Comparing knowledge-based event extraction methods with data-driven methods generates several insights. In contrast to data-driven methods, knowledge-based techniques require little data, but conversely, they need a considerable amount of linguistic, lexicographic, and – for lexico-semantic patterns – also domain knowledge, inherently increasing the amount of required expertise. Compared to data-driven extraction methods, the emphasis is less on training (development) times, but more on execution times (needed for detecting all the required concepts). Moreover, among the knowledge-based methods, there are several differences. Lexico-syntactic patterns (K_1 to K_5) require less data for initial training (clustering) phases, as documents have more useful contents, yet lexico-semantic patterns (K_6 to K_{12}) often require more development time. The results of lexico-semantic patterns additionally excel in interpretability and quality due to their traceability, yet maintenance costs are considerably higher than for lexico-syntactic patterns.

2.2.3 Hybrid Event Extraction

Research has shown that it is hard to solely apply pattern-based algorithms successfully. These algorithms often need to be bootstrapped or require initial clustering, which can be done by means of statistics. For instance, Piskorski et al. (2007) (point H_1 in Figure 2.2) apply an initial clustering procedure to their data set of online news articles before acquiring extraction patterns using a semi-supervised machine learning approach. Also, results from knowledge-based approaches are often used in subsequent, data-driven processing steps, e.g., filtering discovered events using term occurrence statistics (Chun et al., 2004) (H_2). Alternatively, methods that traditionally have been statistics-based, such as part-of-speech tagging, can be improved by adding domain knowledge, which is especially useful in the biomedical domain where common words are often used differently, but also in many other domain-specific and language-specific cases (Lee et al., 2003) (H_3). Moreover, in a similar fashion as presented by Piskorski et al. (2007), knowledge-based event extraction patterns can be learned by applying machine learning techniques, such as conditional random fields, and support vector machines (Best et al., 2008; Björne et al., 2010; Jungermann and Morik, 2008; Tran et al., 2012) (labeled H_4 to H_7 , respectively). In such approaches, patterns are iteratively constructed (semi-)randomly using their basic (conjunction, disjunction, and negation) operators and (lexical, syntactic, and semantic)

elements. The generated patterns are optimized using various criteria which are usually based on the maximization of the F_1 scores on a specific annotated test set.

In hybrid event extraction systems, due to the usage of data-driven methods, the amount of required data increases with respect to knowledge-driven systems, yet typically remains less than is the case with purely data-driven methods. Compared to a knowledge-driven approach, complexity – and hence required expertise – is generally high as well due to the combination of multiple techniques. This also leads to higher training and possibly higher execution times. On the other hand, the amount of expert knowledge that is needed for effective and efficient event discovery is often less than for pattern-based methods, because lack of domain knowledge can be compensated for by using statistical methods. As for the interpretability, attributing results to specific parts of the event extraction is more difficult due to the addition of data-driven methods. Yet, interpretability still benefits to some extent from the use of semantics as in knowledge-based approaches.

2.3 Applications

The applications of event extraction are very diverse, and can be divided into two major fields. First, event extraction has a wide range of utilizations in the biomedical domain (Björne et al., 2010; Chun et al., 2004; Cohen et al., 2009; Miwa et al., 2010; Riedel et al., 2009; van Landeghem et al., 2013; Yakushiji et al., 2001), for instance for identifying molecular events, protein bindings, and gene expressions, which can subsequently be used in biomedical research. Figure 2.3 is a typical example of such tools, and depicts the graph-based EVEX user interface for browsing large-scale databases for biomedical events discovered in PubMed abstracts and full-text articles (van Landeghem et al., 2013). Here, the nodes represent biological entities like proteins, which are interconnected through edges, representing relations or events. Upon selecting one of these events, many associated properties, e.g., type, polarity, or extraction confidence, are retrieved.

Second, many applications of event extraction can be distinguished in news digestion tasks. Usually, event extraction is performed for summarization purposes (Lee et al., 2003) to compress large news messages into a small number of auto-generated sentences based on identified events, but it has also proven to be useful in news personalization systems (Borsje et al., 2010) for selecting relevant news items with respect to events that have an associated user preference. Furthermore, news event applications are found in algorithmic trading (Nuij et al., 2014), risk analysis (Capet et al., 2008), and decision making support tools (Wei and Lee, 2004), where the identified events are often transformed into numerical or binary signals, based on which actions are undertaken.

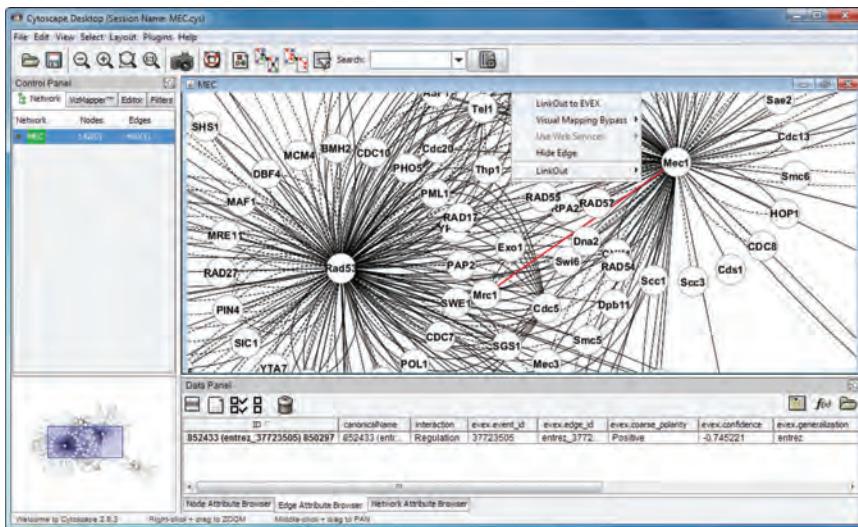


Figure 2.3: EVEX user interface for browsing large-scale databases for biomedical events.

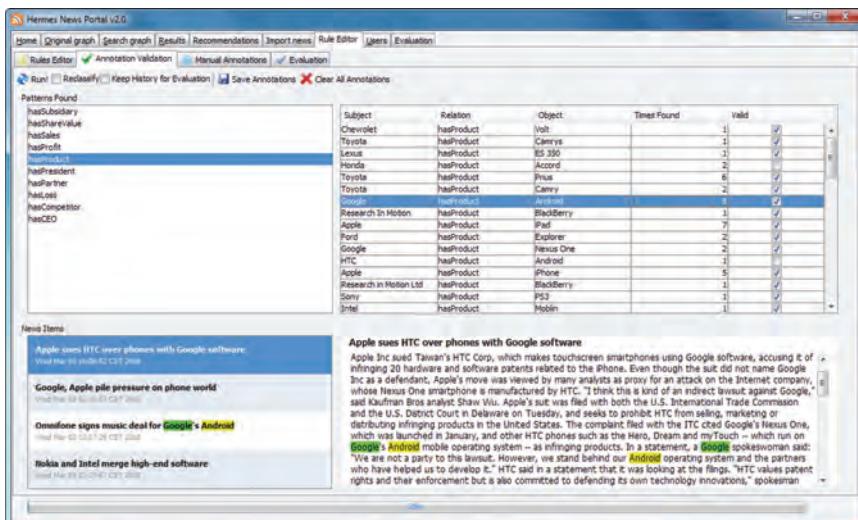


Figure 2.4: Hermes user interface for browsing news feeds for financial events.

Most news-oriented event extraction applications are aimed at general news processing (Aone and Ramos-Santacruz, 2000; Lee et al., 2003; Lei et al., 2005; Liu et al., 2008; Naughton et al., 2006; Tran et al., 2012), but event extraction has also been applied to process scientific (Vargas-Vera and Celjuska, 2004) and award-related news (Xu et al., 2006). The financial domain is yet another popular application area of news event extraction (Arendarenko and Kakkonen, 2012; IJntema et al., 2012; Li et al., 2002), of which an example, the Hermes News Portal (IJntema et al., 2012), is displayed in Figure 2.4. Here, financial events are shown to brokers, in order to assist them in daily trading tasks. Events are extracted based on user-defined lexico-semantic patterns, and are displayed to the user for approval. Also, since the 1980s there has been a great demand for event-based solutions for security-related topics such as terrorism, armed conflicts, and epidemiology, which still generates new research outputs today (Atkinson et al., 2009, 2013; Piskorski et al., 2007; Tanev et al., 2008).

Other applications found in recent literature are event discovery in political documents (Jungermann and Morik, 2008), legal documents (Lagos et al., 2010), historical archives (Cybulska and Vossen, 2011), and blogs (Nishihara et al., 2009; Okamoto and Kikuchi, 2009). Some recent approaches do not limit themselves to written text from documents and streams, but even consider television broadcasts and videos (Chen et al., 2007) for news summarization and security applications. For this purpose, transcripts are used, but more recently, research also moves towards image processing, e.g., for monitoring systems (Kamijo et al., 2000).

2.4 Research Issues

In event extraction, there are many open research issues and points of particular interest, of which the main ones are related to:

- the context-based favourability of data-driven, knowledge-driven, or hybrid approaches;
- understanding the limitations of specific event extraction techniques;
- the domain-dependency of event extraction procedures, affecting both their flexibility and effectiveness;
- the scalability of event extraction approaches when dealing with big data;
- and the complexity of extracted events.

Similar to what can be observed for the field of IE, there is an ongoing debate on the superiority of data-driven and knowledge-driven approaches to event extraction. Although for both approaches more or less equal performances have been reported in literature, advocates of data-driven techniques emphasize their favourable (real-time) computability, whereas knowledge-driven approaches are advocating a higher degree of interpretability due to the general traceability of the results. Users of hybrid event extraction approaches, on the other hand, effectively combine both approaches to their advantage. To determine the best technique for specific applications, there is a need for further research into the best scenarios for the successful application of each technique.

Also, depending on the application at hand, it is important to understand the limitations (and hence the suitability) of the employed techniques. For instance, opting for knowledge-driven approaches when the quality of annotations cannot be ensured or assessed properly, is generally a bad idea, and data-driven approaches should be considered instead. Conversely, when the amount of available data is sparse, most data-driven approaches will give results, yet their correctness and reliability is debatable and subject to further evaluation. Furthermore, in the biomedical domain, regular part-of-speech parsers are less useful than in other applications such as (financial) news processing, as there are many special terms, but also common words are used differently and with different meanings, requiring retrained, specialized parsers instead. This is definitely not an isolated case. At the moment, many researchers acknowledge the existence of such issues in many domains, yet there is little research on the identification of and principle solutions to such issues.

Generally, event extraction is a closed domain procedure. This means that moving to new domains requires retraining data-driven methods and reformulating patterns used for knowledge-driven event extraction, reducing the flexibility of most state-of-the-art approaches to event extraction. Scaling up to larger data sets inherently involves increasing processing power and memory to hold, update, and reason with trained models or knowledge bases. In order to cope with such problems, as well as with additional issues related to the real-time application of event extraction approaches, research taking into account big data phenomena is necessary (Baeza-Yates, 2013). Such research alleviates processing issues associated with large data sets and multi-domain or general knowledge bases, by accounting for scalability issues. For this, researchers aim to optimize algorithms for big data environments, or develop methodologies for parallel and distributed settings.

However, domain and scale changes affect pattern-based approaches in more ways than can be accounted for with solutions borrowed from big data research, revealing additional research issues specifically for knowledge-driven approaches. As the knowledge

required for patterns is often substantial, such changes drive up the costs for acquiring and maintaining patterns, and additionally increase the danger of consistency errors. Moreover, because humans use natural language in their own unique way, there are many different ways of describing similar information. Patterns have to be highly flexible so that they fit many variants, without compromising their accuracy. Therefore, patterns need to be defined in such a way that they match as many events as possible in an arbitrary corpus (i.e., a high recall), without cutting down on precision. Up until now, computer-aided pattern learning techniques that aim to alleviate these problems by automating the pattern construction process have not yet resulted in satisfactory results that are comparable to those of hand-crafted patterns. These developments stress the need for research into pattern learning and optimization.

Last, recent event extraction research has primarily focused on extraction procedures and applications of moderate complexity, thereby putting less emphasis on more complex notions of events. For instance, taking into consideration recent developments in sentiment analysis (Feldman, 2013), events can be enriched based on the prevailing sentiment regarding the event itself or its (in)directly related actors, associating discovered facts with a sentiment-based weighting scheme so as to increase the utility of events in for instance decision making processes. Further recent enhancements include the identification of event sequences (Kengne et al., 2013), enabling one to track events over time, and the connection of events to places (e.g., by means of geotags) (Ho et al., 2012). However, fully taking into account spatio-temporal aspects while maintaining efficient and accurate reasoning has proven to be a difficult challenge that stimulates further research.

2.5 Development

In the process of developing event extraction systems, system engineers have a variety of options. First, there are many programs and packages specifically useful for data-driven event extraction, including machine learning tools with graphical user interfaces, which enable easy training of (statistical) models that can serve as a basis for data-driven event extraction systems, without requiring any programming or scripting efforts. This software is ideal for development, but can also be employed to train real models that are fed into other tools. For instance, the Java-based Weka package (<http://cs.waikato.ac.nz/ml/weka/>) offers tools for data pre-processing, classification, regression, clustering, association rules, and visualization. Other useful software distributions are the Java-based RapidMiner tool (<http://rapidminer.com>), formerly known as YALE, the Python-oriented Orange framework (<http://orange.biolab.si/>)

for data visualization and analysis with additional machine learning, bioinformatics, and text mining functionalities, and a graphical user interface for data mining using R, Rattle, which can be found at <http://rattle.togaware.com/>. Furthermore, there are statistical processing tools, such as Matlab and R, which offer useful functionalities for event extraction. For instance, R offers a great many of external packages, each of which are mostly attributed to a specific subset of machine learning techniques. Similarly, Matlab has various toolboxes that offer learning algorithms, such as the Statistics and Neural Network toolboxes.

For most popular programming languages, there is a wide range of excellent packages that can be employed for (data-driven) event extraction. For Java, the API of Weka and the Java-ML library (<http://java-ml.sourceforge.net/>) are commonly used for incorporating machine learning procedures into applications. Additionally, the Apache Mahout project (<http://mahout.apache.org>) provides a Java library for scalable machine learning. Tools for Python are plentiful. PyBrain (<http://pybrain.org/>), for instance, offers support for neural networks, reinforcement learning, unsupervised learning, and evolution, whereas mlpjy (<http://mlpjy.sourceforge.net/>) offers a vast array of regression, classification, clustering, and dimensionality reduction tools. There are also cross-language libraries, such as SHOGUN (<http://shogun-toolbox.org/>), which is implemented in C++ and has interfaces to Matlab, R, Octave, and Python.

Next, there is also a wide range of NLP tools and libraries available for knowledge-driven event extraction, which can also be useful for some data-driven and hybrid event extraction applications. Many tools have been created for specific parsing tasks, e.g., tokenization, part-of-speech tagging, and named entity recognition tools provided by the Stanford NLP group (<http://nlp.stanford.edu/software/>). However, there is an increasing number of packages that support many of such tasks. For instance, for Java programmers, there are frequently used highly configurable IE pipeline libraries, such as GATE (<http://gate.ac.uk/>) and LingPipe (<http://alias-i.com/lingpipe/>), which implement many parsers and supply a convenient plug-and-play pipeline architecture. For Python users, there are comparable packages available, such as NLTK (<http://nltk.org/>).

When incorporating knowledge into event extraction systems, semantics can be added by using a semantic lexicon, which is for instance accessible through the WordNet API (<http://wordnet.princeton.edu/>). Moreover, there are many tools for parsing, storing, querying, and reasoning with semantic data. For Java, one could choose between Jena (<http://jena.apache.org/>), Pellet (<http://clarkparsia.com/pellet/>), OWL API (<http://owlapi.sourceforge.net/index.html>), and OpenRDF Sesame ([http:](http://)

//www.openrdf.org/), to name a few. Also, for other languages such as Python and C/C++ there are many alternatives, e.g., rdflib (<https://github.com/RDFLib>), seth (<http://seth-scripting.sourceforge.net/>), and Redland (<http://librdf.org/>).

2.6 Evaluation

For the evaluation of event extraction methods, researchers often rely on quantitative indicators, measuring performance using a golden standard-based approach. Data sets, consisting of news messages, documents, articles, etc., are annotated by domain experts, meticulously detailing the events that should be found by the (semi-)automatic event extraction approaches. We distinguish various levels of annotation, ranging from coarse-grained to fine-grained specifications of events. Most annotations are done on a document level, some annotations focus on paragraphs, and a minority of the available annotations cover all sentences of a text segment. Moreover, there are differences in the level of detail in event annotations. Most event annotations merely collect the event itself with a subject and (if applicable) an object (e.g., the Hermes News Portal for financial events displayed in Figure 2.4), whereas others provide well-decorated events, describing many properties (e.g., the EVEX user interface for biomedical events in Figure 2.3). The granularity of the annotations, as well as their level of detail, directly affect the costs of data processing, with coarse-grained, low-detailed data sets being more cost-effective than fine-grained, high-detailed data sets. For their applicability and evaluative value, the opposite is true.

In accordance with IE and TM, performance is generally measured by computing the number of true positives and negatives, as well as the number of false positives and negatives, each of which can be determined using a golden standard data set, composed by domain experts using a minimum Inter-Annotator Agreement (typically between 60% and 90%, depending on the number of annotators) in order to improve data quality. Based on these numbers, precision (fraction of retrieved events that are relevant), recall (fraction of relevant events that are retrieved), and their harmonic mean, the F_1 score, are computed. In general, F_1 scores around 70% are considered normal, and anything above 90% is perceived as exceptionally good. For some highly ambiguous domains, these numbers are lower. Also, for specific applications, F_1 scores are not considered, and researchers solely focus on improving either precision or recall, depending on their needs.

Pre-annotated data sets for event extraction are still rather scarce, as manual annotation is a costly process. The BioNLP'09 shared task on event extraction offers an annotated set at <http://www.nactem.ac.uk/tsujii/GENIA/SharedTask>. The set is based on the GENIA corpus (a semantically annotated biomedical corpus), and is potentially use-

ful for benchmarking purposes, as 24 teams have reported their final results at the time, and many afterwards. Alternatively, there is a small corpus on general events available at <http://www.isi.edu/~hobbs/EventDuration/annotations/>. Moreover, the DeRiVE workshops, organized in conjunction with the ISWC conferences, provide large, general purpose data sets. In the 2011 challenge, a large data set with music and entertainment events is provided at <http://semanticweb.cs.vu.nl/derive2011/Challenge.html>. In the 2013 data challenge (<http://derive2013.wordpress.com/data-challenge/>), the focus shifted to linked open data, inviting participants to combine sensor data (expressed in parsable messages) with maritime data sets on vessels, smuggling, pollution, etc. In the next few years, we expect to see more of such initiatives, resulting in a future defacto benchmarking standard for event extraction, similar to today's TREC (<http://trec.nist.gov>) challenges for general IE.

The reusability of existing data sets such as the latter examples greatly depends on the targeted domains. Therefore, in practice, data are usually scraped from (news) feeds at Reuters, Bloomberg, Yahoo!, etc., after which they are filtered and annotated by domain experts. Crowdsourcing solutions, e.g., services such as Amazon Mechanical Turk (<http://aws.amazon.com mturk/>) and CrowdFlower (<http://crowdflower.com/>), are great alternatives for obtaining annotations, as they are fast, cheap, and have access to a large pool of potential annotators (Wichmann et al., 2011). Although annotators are usually not domain experts, inconsistencies and inaccuracies can be overcome by using basic measures such as the inter-annotator agreement. Moreover, crowdsourcing services often aid in identifying fraudulent users, and additionally allow for qualification tests in order to further increase annotation quality (Hsueh et al., 2009; Ipeirotis et al., 2010; Kern et al., 2010).

2.7 Conclusions

Event extraction has gained in popularity due to its wide applicability for various purposes. In this chapter, we reviewed the various data-driven, knowledge-driven, and hybrid techniques of event extraction, and evaluated the works on a set of qualitative dimensions, i.e., the amount of required data, knowledge, and expertise, as well as the interpretability of the results and the required development and execution time. We identified the major strengths and weaknesses of the main event extraction techniques, as well as their major differences. Data-driven approaches require a lot of data available and little domain expertise, while knowledge-driven approaches work adequately on small data sets, but require more expert knowledge. Hybrid methods inherit the benefits of both data-

driven and knowledge-driven approaches, mitigating their disadvantages. Moreover, we discussed the main application areas of event extraction, e.g., the biomedical, security, and financial domains. We additionally identified several research issues that need to be addressed, such as approach scalability and domain dependencies. Last, we provided pointers to tools and libraries for developing event extraction systems, and discussed the evaluation of event extraction systems.

In the near future, we envisage event extraction to evolve in various ways. First, current encouraging developments in sentiment analysis (Feldman, 2013) can stimulate event extraction research by connecting sentiment to events, which are currently often merely rich facts decorated with actors and other properties. Also, as the current field is already moving toward identifying event sequences (Kengne et al., 2013), tracking events over time (Verheij et al., 2012a), and connecting events to places (e.g., by means of geotags) (Ho et al., 2012), a next possible step in event extraction could be fully taking into account spatio-temporal aspects, not only by connecting these aspects to events, but also by exploiting this information in reasoning and discovering new events. Furthermore, we envisage better real-time performances due to improved hardware and the rise of computing clusters, but also due to the output of current and ongoing research into big data, resulting in scalable solutions. Last, with the advent of linked open data and large accessible knowledge bases such as DBpedia, many more application domains can be supported, reducing the need for creating and maintaining gazetteering lists and ontologies for knowledge-based event extraction techniques. The latter developments will make event extraction more accessible and trustworthy, facilitating the development of previously unenvisioned applications of event extraction.

Chapter 3

A Semantics-Based Event Extraction Framework[†]

ACCURATE and timely automatic identification of events in news items is crucial in today’s financial markets, as they are sensitive to breaking news on financial events. Unstructured news items originating from many heterogeneous sources have to be mined in order to extract knowledge useful for guiding decision making processes. Hence, we propose the Semantics-Based Pipeline for Economic Event Detection (SPEED), focusing on extracting financial events from news articles and annotating these with meta-data at a speed that enables real-time use. In our implementation, we use some components of an existing framework as well as new components, e.g., a high-performance Ontology Gazetteer, a Word Group Look-Up component, a Word Sense Disambiguator, and components for detecting financial events. Through their interaction with a domain-specific ontology, our novel, semantically-enabled components constitute a feedback loop which fosters future reuse of acquired knowledge in the event detection process.

[†]This chapter is based on the article “A. Hogenboom, F. Hogenboom, F. Frasincar, K. Schouten, and O. van der Meer. Semantics-Based Information Extraction for Detecting Economic Events. *Multimedia Tools and Applications*, 64(1):27–52, 2013.”

3.1 Introduction

Communication plays an important role in today's society, as it provides ways to convey messages, typically with a specific goal in mind. Communication can thus facilitate effective, well-informed decision making. Recent decades have shown a tendency of human communication to expand – driven by the increasing popularity of automating processes – such that it also includes human-machine interaction besides purely human interaction. So far, communication between humans and machines has been thwarted by the disability of machines to fully understand complex natural language. Humans have hence adapted their communication with machines by using clearly defined, fixed, and unambiguous morphology, syntax, and semantics. Yet, this only provides limited means of communication. It is the flexibility and complexity of human language that makes it so expressive. Hence, in order to enable more effective human-machine communication, machines should be able to understand common human language. This is one of the promises of the ongoing research on automated Natural Language Processing (NLP).

In today's information-driven society, machines that can process natural language can be of invaluable importance. Decision makers are expected to process a continuous flow of (news) messages or any kind of raw data through various input channels, by extracting information and understanding its meaning. Knowledge can then be acquired by applying reasoning to the gathered information. However, the amount of available data is overwhelming, whereas decision makers need a complete overview of their environment in order to enable effective, well-informed decision making. In today's global economy, this is of paramount importance. Decision makers need an intuition on the state of their market, which is often extremely sensitive to breaking news on financial events like acquisitions, stock splits, or dividend announcements. In this context, the identification of events can guide decision making processes, as these events provide means of structuring information using concepts, with which knowledge can be generated by applying inference. Automating information extraction and knowledge acquisition processes can facilitate or support decision makers in fulfilling their cumbersome tasks, as faster processing of more data enables one to make better informed decisions.

Therefore, we aim to have a fully automated application for processing financial news messages – fetched from Really Simple Syndication (RSS) (Winer, 2003) feeds – in such a way that the essence of the messages is extracted and captured in events that are represented in a machine-understandable way. Thus, in line with the philosophy of the Semantic Web (Berners-Lee et al., 2001), the extracted events can be made accessible for other applications as well, e.g., in order to enable knowledge acquisition. Furthermore,

the application should be able to handle news messages at a speed that is sufficient for real-time use, because new events can occur any time and require decision makers to respond in a timely and adequate manner.

We propose a framework (pipeline) that identifies the concepts of interest (i.e., concepts related to financial events), which are defined in a domain ontology and are associated to synsets from a semantic lexicon (WordNet (Fellbaum, 1998)). A preliminary version of this Semantics-based Pipeline for Economic Event Detection (SPEED) has been proposed by Hogenboom et al. (2010d). In our current endeavors, we elaborate on this framework by providing a more extensive discussion of the specifics of our framework (e.g., its components and algorithms), as well as a more detailed (component-wise) performance evaluation. For concept identification, we match lexical representations of concepts retrieved from the text with event-related concepts that are available in WordNet, and thus aim to maximize recall. Here, we use lexico-semantic patterns based on concepts from the ontology. The identified lexical representations of relevant concepts are subject to a procedure for identifying word groups rather than individual words as well as a word sense disambiguation procedure for determining the corresponding sense, in order to maximize precision. In order to support real-time applicability, we also aim to minimize the latency, i.e., the processing time of a news message.

Our contributions are two-fold. The first contribution relates to our proposed combination of a number of existing techniques and a number of new components into a novel pipeline for event extraction. As our pipeline is semantically enabled, it is designed to generalize well to other domains, which would typically require the existing ontology to be replaced by other domain-specific ones. Through their interaction with a domain-specific ontology, our novel, semantically-enabled components constitute a feedback loop which fosters future reuse of acquired knowledge in the event detection process. An additional contribution lies in the efficiency and effectiveness of our newly proposed components for identifying relevant ontology concepts, word group look-up, and word sense disambiguation. Our framework, which also builds on previous work on news personalization (Borsje et al., 2008; Schouten et al., 2010), distinguishes itself by means of its fast ontology gazetteer, precise discovery of events using word sense disambiguation, and event decoration with related information using lexico-semantic patterns (Borsje et al., 2010).

This chapter is structured as follows. First, Section 3.2 discusses related work. Subsequently, Section 3.3 elaborates on the proposed framework and its implementation. The approach is evaluated in Section 3.4. Last, Section 3.5 concludes the chapter and provides directions for future research.

3.2 Related Work

This section discusses tools that can be used for Information Extraction (IE) purposes. First, we elaborate on SemNews, which is an application that aims at accurately extracting information from heterogeneous news sources. Then, we continue by focusing on IE pipelines.

3.2.1 SemNews

SemNews (Java et al., 2006) is a Semantic Web-based application that aims to discover the meaning of news items. These items are retrieved from RSS feeds and are processed by the NLP engine OntoSem (Nirenburg and Raskin, 2001). The engine retrieves Text Meaning Representations (TMR), which are subsequently stored in an ontology (fact repository) that holds as a representation of the world. Results are then published in Ontology Web Language (OWL) (Bechhofer et al., 2004) format, so that they can be used in Semantic Web applications. This approach is very much related to the work of Vargas-Vera and Celjuska (2004), as they present an approach to recognize events in news stories and to populate an ontology semi-automatically.

The information extraction process of OntoSem can be divided into several stages that the application goes through for each news article that is to be analyzed. First, the Preprocessor ensures that sentence and word boundaries are identified, as well as named entities, acronyms, numbers, dates, etc. Then, the Syntactic Parser is invoked to analyze the syntax of the corpus and to resolve syntactic ambiguity. The parsed text is passed through the Basic Semantic Analyzer, which produces a basic TMR using various concepts defined in the ontology and copes with resolving semantic ambiguity. Subsequently, there is a phase that is associated with extended analysis, such as resolving referential ambiguity and temporal ordering. Finally, the fact repository is updated by the Fact Extractor, using the knowledge stored within the extended TMR.

SemNews seems to suit the approach we aim for well. However, OntoSem employs a frame-based language for representing the ontology and an onomasticon for storing proper names, whereas we envisage an approach in which both the input ontology and the facts extracted from news items are represented in OWL, as this fosters application interoperability and the reuse of existing reasoning tools. Also, the use of an onomasticon is not sufficient when disambiguating word senses, and hence a general semantic lexicon like WordNet is desired.

3.2.2 ANNIE

Most IE-focused tools utilize their own framework for information extraction. However, over the last few years, GATE (Cunningham, 2002; Cunningham et al., 2002), a freely available general purpose framework for IE purposes, has become increasingly popular as a basis for IE tools. GATE is highly flexible in that the user can construct natural language processing pipelines from components that perform specific tasks. One can distinguish between various linguistic analysis applications such as tokenization (e.g., distinguishing words), syntactic analysis jobs like Part-Of-Speech (POS) tagging, and semantic analysis tasks such as understanding. By default, GATE loads the A Nearly-New Information Extraction (ANNIE) system, consisting of several key components which can be useful components for many custom natural language processing pipelines.

The first component in the ANNIE pipeline is the English Tokenizer, which splits text into separate chunks, such as words and numbers, and takes into account punctuation. The tokenizer is a vital component and other components rely upon its output. The next component is the Sentence Splitter, which splits text into sentences. Subsequently, the POS Tagger determines the part-of-speech (e.g., noun, verb, etc.) of words within a scanned corpus. The fourth component in the ANNIE pipeline is the Gazetteer, which identifies named entities in the corpus that is processed, such as people, organizations, percentages, etc. After defining named entities and after annotating words with their proper POS tags, there could be a need to combine and disambiguate discovered annotations. The fifth component in ANNIE, i.e., the NE (Named Entity) Transducer, employs JAPE rules, which only offer limited support to express in a generic way rules geared towards for example combining and disambiguating entities. Finally the last component, the OrthoMatcher, adds identity relations between named entities found earlier in the pipeline. Its output can for instance be used for orthographic co-referencing, which is not part of ANNIE.

There are several tools or frameworks that utilize the ANNIE pipeline, or use (modified) ANNIE components together with newly developed components. For instance, Artequakt (Kim et al., 2002) aims to generate tailored narrative artist biographies using automatically annotated articles from the Web. In their semantic analysis, they employ GATE components for gazetteering and named entity recognition. Another example of a tool that uses ANNIE components is Hermes (Borsje et al., 2008), which extracts a set of news items related to specific concepts of interest. For this purpose, semantically enhanced ANNIE GATE components are used, i.e., they make use of concepts and relations stored in ontologies.

Although the ANNIE pipeline has proven to be useful in various information extraction jobs, its functionality does not suffice when applied to discovering financial events in news messages. For instance, ANNIE lacks important features such as a component that focuses on performing Word Sense Disambiguation (WSD), although some disambiguation can be done using JAPE rules in the *NE Transducer*. This is however a cumbersome and ineffective approach where rules have to be created manually for each term, which is prone to errors. Furthermore, ANNIE lacks the ability to individually look up concepts from a large ontology within a limited amount of time. Nevertheless, GATE is highly flexible and customizable, and therefore ANNIE's components are either usable, or extendible and replaceable in order to suit our needs.

3.2.3 CAFETIERE

Besides the Artequakt and Hermes frameworks, another example of an adapted ANNIE pipeline is the Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RElations (CAFETIERE) relation extraction pipeline (Black et al., 2005), developed in the Parmenides project (Mikroyannidis et al., 2005; Rinaldi et al., 2003). The pipeline contains an ontology lookup process and a rule engine. Within CAFETIERE, the Common Annotation Scheme (CAS) DTD is applied, allowing for three annotation layers, i.e., structural, lexical, and semantic annotation. CAFETIERE employs extraction rules defined at lexico-semantic level which are similar to JAPE rules. Nevertheless, the syntax is at a higher level than is the case with JAPE, resulting in easier to express, but less flexible rules.

Because CAFETIERE stores knowledge in an ontology by means of the Narrative Knowledge Representation Language (NKRL), Semantic Web ontologies are not employed. NKRL has no formal semantics and lacks reasoning support, which is desired when identifying for instance financial events. Furthermore, gazetteering is a slow process when going through large ontologies. Finally, the pipeline also misses a WSD component.

3.2.4 KIM

The Knowledge and Information Management (KIM) platform (Popov et al., 2004a) provides another infrastructure for IE purposes, by combining the GATE architecture with semantic annotation techniques. The back-end and middle layer of the KIM platform focus on automatic annotation of news articles, where named entities, inter-entity relations, and attributes are discovered. For this, it is employed a pre-populated OWL upper ontology, i.e., a minimal but sufficient ontology that is suitable for open domain and gen-

eral purpose annotation tasks. The semantic annotations in articles allow for applications such as semantic querying and exploring the semantic repository.

KIM’s architecture is a conglomeration of three layers. In the back-end, a standard GATE pipeline is invoked for named entity recognition with respect to the KIM ontology. The GATE pipeline is altered in such a way that its components are semantically enabled, and is extended with semantic gazetteers and pattern-matching grammars. Furthermore, GATE is used for managing the content and annotations within the back-end of KIM’s architecture. The middle layer of the KIM architecture provides services that can be used by the topmost layer, e.g., semantic repository navigation, semantic indexing and retrieval, etc. The topmost layer of KIM embodies front-end applications, such as the Annotation Server and the News Collector.

The differences between KIM and our envisaged approach are in that we aim for a financial event-focused information extraction pipeline, which is in contrast to KIM’s general purpose framework. Hence, we employ a domain-specific ontology instead of an upper ontology. Also, we specifically focus on extracting events from corpora, and not on (semantic) annotation. Furthermore, no mention has been made regarding WSD within the KIM platform, whereas we consider WSD to be an essential component in an IE pipeline.

3.2.5 Discussion

Although the approaches to information extraction we discussed so far each have their advantages, they also fail to address some of the issues we aim to alleviate. From a technical point of view, the frameworks incorporate semantics only to a limited extent, which is also demonstrated by Table 3.1. For instance, they make use of gazetteers or knowledge bases that either do not use ontologies or employ ontologies that are not based on OWL and thus do not make use of existing standards to represent ontologies. Being able to use a standard language as OWL fosters application interoperability and the reuse of existing reasoning tools. Also, to the best of our knowledge, existing applications typically lack a feedback loop, i.e., the acquired knowledge is not used for future information extraction. Furthermore, WSD is absent and the focus often is on annotation, instead of event recognition. Therefore, we aim for a framework that combines the insights gained from the approaches that are previously discussed, targeted at the discovery of financial events in news articles.

Approach	Purpose	Input	Output	KB utilization	KBΔ	WSD
SemNews	Extraction of facts	RSS	OWL ontology	Onomasticon for proper names and frame-based language	No	No
ANNIE	Detection of entities	Text	Annotations, XML	Gazetteering entity lists	No	No
CAFETIERE	Detection of relations and entities	Text	Annotations, XML	Gazetteering NKRL ontology	No	No
KIM	Detection of entities	Text	Annotations, RDF(s) ontology	Gazetteering RDF(s) ontology	Yes	No
Desired	Detection of financial events !	RSS	Annotations, OWL ontology	Reasoning with OWL ontology and a general semantic lexicon	Yes	Yes

Table 3.1: Comparison of existing approaches and the characteristics required for our current endeavors, based on purpose (*Purpose*), input (*Input*), output (*Output*), knowledge base utilization (*KB utilization*), presence of knowledge base updates (*KBΔ*), and usage of word sense disambiguation (*WSD*).

3.3 Financial Event Detection based on Semantics

The analysis presented in Section 3.2 demonstrates several approaches to automated information extraction from news messages. However, the state-of-the-art in text processing does not enable us to perform the specific task we aim to perform. Current approaches are more focused on annotation of documents, whereas we strive to actually extract information – specific financial events and their related concepts – from documents, with which, e.g., a knowledge base can be updated.

In order to be able to discover financial events in written text, the analysis of texts needs to be driven by semantics, as the domain-specific information captured in these semantics facilitates detection of relevant concepts. Therefore, we propose the **Semantics-Based Pipeline for Economic Event Detection (SPEED)**, consisting of several components which sequentially process an arbitrary document, as visualized in Figure 3.1. These components are supported by a semantic lexicon (i.e., WordNet) and a domain-specific ontology.

Due to the potential of the General Architecture for Text Engineering (GATE), we use this IE framework for its modularity. However, none of the existing applications of the general GATE architecture can support the tasks we seek to perform. Even more, no

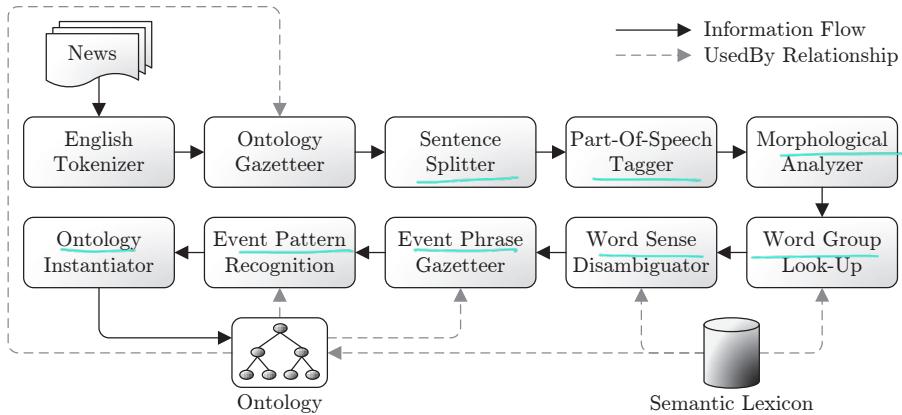


Figure 3.1: SPEED design.

implementation exists of several specialized envisioned components. Therefore, the Java-based implementation of our proposed pipeline requires the development of techniques that support our needs. The default GATE implementations of the English Tokenizer, Sentence Splitter, Part-Of-Speech Tagger, and the Morphological Analyzer suit our needs to a limited yet for now sufficient extent.

This section continues by explaining the domain ontology that supports our pipeline in Section 3.3.1. Subsequently, Sections 3.3.2 through 3.3.11 discuss the pipeline's individual components. We run through the processing steps of the SPEED framework by means of a typical example news item, displayed in Figure 3.2. This short news item was extracted from the Yahoo! Business and Technology newsfeed and discusses Google's acquisition of YouTube. In our pipeline, each individual component adds its own annotations to the example news item above. These annotations can be considered as multiple layers on top of the corpus. This means that one word can have multiple annotations, and can also be part of a larger annotation spanning multiple words at the same time.

SAN FRANCISCO (Reuters) - Web search leader Google Inc. on Monday said it agreed to acquire top video entertainment site YouTube Inc. for \$1.65 billion in stock, putting a lofty new value on consumer-generated media sites.

Figure 3.2: A typical news example.

3.3.1 Domain Ontology

Our envisaged approach is driven by an ontology containing information on the NASDAQ-100 companies, extracted from Yahoo! Finance. This domain ontology has been developed by domain experts through an incremental middle-out approach. The ontology captures concepts and events concerning the financial domain, e.g., companies, competitors, products, CEO's, etc. Many concepts in this ontology stem from a semantic lexicon (i.e., WordNet) and are linked to their semantic lexicon counterparts, but a significant part of the ontology consists of concepts representing named entities (i.e., proper names). In our ontology, we distinguish between ten different financial events, i.e., announcements regarding CEOs, presidents, products, competitors, partners, subsidiaries, share values, revenues, profits, and losses, which are supported by appropriate classes and properties.

We validated our domain ontology using OntoClean (Guarino and Welty, 2002), a methodology for analyzing ontologies that uses notions for philosophical ontological analysis. OntoClean is based on formal, domain-independent class properties (meta-properties and their modifiers), i.e., identity, unity, rigidity, and dependence. Once annotated with these meta-properties, the ontology can be considered to be valid (or “clean”) whenever no constraints are violated that are based on these properties.

3.3.2 English Tokenizer

SPEED is designed to identify relevant concepts and their relations in a document. To this end, first, individual text components are identified as such using the *English Tokenizer*, which splits text into tokens (e.g., words, numbers, or punctuation) and subsequently applies rules specific to the English language in order to split or merge identified tokens. For example, the token combination '| |60| |s| would be merged into one token ‘'60s’. Note that spaces are considered as special tokens and are annotated as a **SpaceToken** rather than a **Token**. For our running example, this translates to the annotations shown in Figure 3.3, where tokens are shaded in medium and light tones (for the sake of clarity) and spaces have a dark shading.

The text is annotated with tokens. Shaded tokens include: SAN, FRANCISCO, (Reuters), - (dark), Web, search, leader, Google, Inc., on, Monday, said, it, agreed, to, acquire, top, video, entertainment, site, YouTube, Inc., for, \$1.65, billion, in, stock, putting, a, lofty, new, value, on, consumer, generated, media, sites. The tokens are shaded in various tones of purple, green, and yellow, while spaces are in black.

Figure 3.3: English Tokenizer annotations (tokens).

3.3.3 Ontology Gazetteer

A first step towards understanding the text is subsequently taken by the *Ontology Gazetteer*, which links concepts in the text to concepts defined in an ontology with relevant concepts (which tend to refer to proper names rather than common words from the semantic lexicon). A normal gazetteer uses lists of words as input, whereas our ontology gazetteer is ontology-driven and scans the text for lexical representations of concepts from the ontology. Matching tokens in the text are annotated with a reference to their associated concepts defined in the ontology. For example, suppose our ontology contains a concept **Google** of type **Company**, with a lexical representation ‘*Google Inc.*’. Any matching ‘*Google Inc.*’ occurrence in the text is then annotated with the concept **Google**.

The default GATE *OntoGazetteer* uses a linear search algorithm to match lexical representations in a text with a list of ontology concepts and their associated lexical representations. However, in our novel *OntoLookup* approach, we use a look-up tree of approximately 5,000 nodes (based on the Yahoo! Finance news messages represented in the ontology), in which possible lexical representations of all relevant concepts in the ontology are mapped to their associated concepts. Each concept can have multiple lexical representations (groups of 1 or more words). These word groups are all represented in the look-up tree. Nodes in the tree represent individual tokens and a path from the root node to an arbitrary leaf node represents a word group.

Figure 3.4 depicts a sample tree structure. In this sample, the root node contains – among other things – references to ‘*Cisco*’, ‘*Google*’, and ‘*Yahoo!*’. The ‘*Cisco*’ token contains a reference to ‘*Systems*’, which in turn contains a reference to a resource in the ontology, as well as to another token, ‘*Inc*’. The latter token also contains a reference to a resource in the ontology, but does not contain a reference to another token. Thus, ‘*Cisco Systems*’, and ‘*Cisco Systems Inc*’ refer to a concept in the ontology. The paths for ‘*Google*’ and ‘*Yahoo!*’ are not fully depicted in Figure 3.4, but could exhibit similar characteristics.

For a given series of tokens, the *OntoLookup* process iterates over the tokens. For each token, it checks whether the look-up tree contains the token. This look-up process starts at the root node of the tree. If the token is not found, the next token in the text is looked up in the root node of the full look-up tree. However, if the token is found, the next token in the text is looked up in the root node of the subtree belonging to the former token. This process is iterated until either a leaf node is reached (i.e., the word group cannot be further expanded), or the root node of the considered subtree does not contain a reference to the next token in the text. The word group associated with the followed path is then annotated with the associated concept from the ontology. The tree

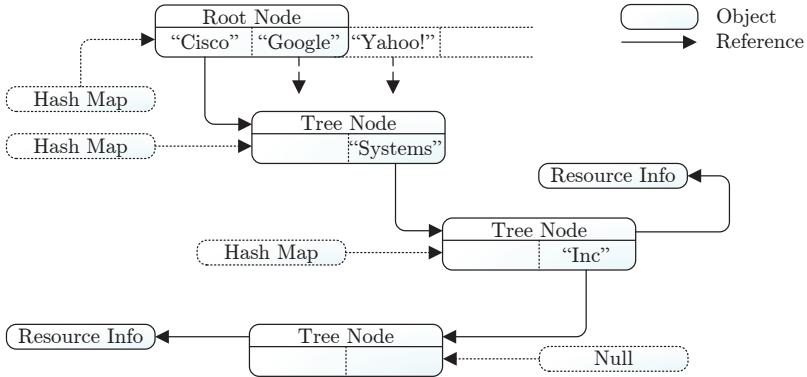


Figure 3.4: Sample *OntoLookup* tree structure.

is implemented using hash maps, in order to reduce the time needed to traverse the tree. The tree structure representing lexical representations of the concepts in our ontology, indexed using hash maps, is of benefit because matching a token with a child node by using, e.g., a linear search algorithm assessing every child node for a possible match with the token is typically less efficient than determining the index of a child node associated with a token by means of hashing.

When run through the discussed component, several concepts are recognized in our running example. As the text is about two companies, i.e., Google and YouTube, the strings referring to these companies are annotated. These lexical representations are stored within the ontology and are linked to the ontology concepts of the type Company, which causes the strings to be annotated with ontology concepts Google and YouTube. Figure 3.5 demonstrates this annotation process, where the highlighted text is annotated with the appropriate ontology concepts.

SAN FRANCISCO (Reuters) - Web search leader Google Inc. on Monday said it agreed to acquire top video entertainment site YouTube Inc. for \$1.65 billion in stock, putting a lofty new value on consumer-generated media sites.

Figure 3.5: *Ontology Gazetteer* annotations (concepts).

3.3.4 Sentence Splitter

Then, the Sentence Splitter groups the tokens in the text into sentences, based on tokens indicating a separation between sentences, which can be, for instance, (a combination of) punctuation symbols or new line characters. The grammatical structure of the text is then uncovered in order to facilitate an initial model of the text's meaning.

As shown in Figure 3.6, grouping tokens into sentences is anything but a straightforward task, as periods do not always denote the end of a sentence, but can also be used as for example decimal separators (or in some languages as thousands separators), in abbreviations, etc. In the case of our leading example, the *Sentence Splitter* fails to find the correct sentences because of the usage of full stops after '*Inc*'. Later on, this is fixed due to the fact that '.' is part of the lexical representation of a concept. Note that the period inside the value of '1.65' is correctly ignored as a full stop.

3.3.5 Part-Of-Speech Tagger

For each sentence, the type of each word token is subsequently determined by the *Part-Of-Speech Tagger*, which tags each word with its part-of-speech. When employing the Part-Of-Speech Tagger, no new annotations are added to the document. Instead, features of tokens are added. Tokens already contain information added by the English Tokenizer on start and end character number, kind (e.g., word, symbol, etc.), length, orthographic category (e.g., lowercase), and the string of characters belonging to the tag. The *Part-Of-Speech Tagger* determines the syntactic category of each token and stores this in a POS feature, which is encoded in capitalized abbreviations. For instance, syntactic categories with suffix VB are verbs, e.g., VZ denotes a verb in third person singular present. Categories beginning with NN are nouns, such as a single proper noun (NNP). Common syntactic categories are displayed in Table 3.2.

SAN FRANCISCO (Reuters) - Web search leader Google Inc. on Monday said it agreed to acquire top video entertainment site YouTube Inc. for \$1.65 billion in stock, putting a lofty new value on consumer-generated media sites.

Figure 3.6: *Sentence Splitter* annotations (sentences).

Category	Description
CC	Coordinating conjunction
CD	Cardinal number
IN	Preposition
JJ	Adjective
NN	Noun
NNP	Proper Noun
PP	Pronoun
RB	Adverb
UH	Interjection
VB	Verb, base form
VBZ	Verb, third person singular present

Table 3.2: Common syntactic categories.

3.3.6 Morphological Analyzer

Different forms of a word have a similar meaning; they relate to the same concept, albeit from possibly different perspectives. Therefore, the *Morphological Analyzer* component subsequently reduces the tagged words to their lemma (i.e., canonical form) and when needed a suffix and/or affix denoting the deviation from this lemma. For instance, for the verb ‘walk’, the ‘walks’ morph is annotated as `root=walk, suffix=s`. Similar to the *Part-Of-Speech Tagger*, the *Morphological Analyzer* does not add new annotations to the document, but token features. When applicable, the *Morphological Analyzer* adds features related to morphology (such as affixes) to the tokens. At any rate, for each token, the root (lemma) is added.

3.3.7 Word Group Look-Up

Words and meanings, denoted often as synsets (set of synonyms) have a many-to-many relationship. A word can have multiple meanings and a meaning can be represented by multiple words. Hence, the next step in interpreting a text is disambiguation of the meaning of the words, given their POS tags, lemmas, etc. To this end, first of all, the *Word Group Look-Up* component combines words into maximal word groups, i.e., it aims at assigning as many words as possible to a group representing some concept in a semantic lexicon such as *WordNet*. We use the complete list of approximately 65,000 existing word groups extracted from *WordNet*. These word groups are represented in a tree structure, where nodes represent individual tokens and a path from the root node to an arbitrary leaf node represents a word group.

SAN FRANCISCO (Reuters) - Web search leader Google Inc. on Monday said it agreed to acquire top video entertainment site YouTube Inc. for \$1.65 billion in stock, putting a lofty new value on consumer-generated media sites.

Figure 3.7: *Word Group Look-Up* annotations (tokens).

Similarly to the *OntoLookup* process, the word group tree can then be used for matching word groups in the text with word groups extracted from the semantic lexicon. For each set of tokens, the tree is traversed until either a leaf node is reached, or the next token in the text is not in the considered subtree. Again, indexing of the tree is implemented using hash maps, in order to optimize the time needed for traversing the tree in the look-up process.

In our running example, where the feature set of the tokens has been previously extended by the *Part-Of-Speech Tagger* and the *Morphological Analyzer*, the *Word Group Look-Up* module of our pipeline employs the *WordNet semantic lexicon* in order to identify word groups, such as ‘*SAN FRANCISCO*’. In Figure 3.7, the tokens in the text that form a word group are merged into a single token.

3.3.8 Word Sense Disambiguator

After identifying word groups, the *Word Sense Disambiguator* determines the word sense of each word group by exploring the mutual relations between senses (as defined in the semantic lexicon and the ontology) of word groups; the stronger the relation with surrounding senses, the more likely a sense matches the context. Grouping words is important, because combinations of words may have very specific meanings compared to the individual words. For instance, ‘*Gross Domestic Product*’ is a combination with a unique meaning that is not associated with any of the individual words in this group. The accuracy of WSD may hence be improved when considering word groups rather than individual words.

We propose an adaptation of the *Structural Semantic Interconnections* (SSI) (Navigli and Velardi, 2005) algorithm for word sense disambiguation. The SSI approach uses graphs to describe word groups and their context (word senses), as derived from a semantic lexicon (e.g., WordNet). The senses are determined based on the number and type of detected semantic interconnections in a labeled directed graph representation of all senses of the considered word groups. Similarities are calculated based on an arbitrary distance measure.

More than other common approaches, the SSI approach enables us to incorporate a notion of semantics into the word sense disambiguation process by exploiting a vast semantic lexical database. Other common approaches are typically restricted to a relatively small collection of representations of ontological concepts (Theobald et al., 2003) or barely use any notion of semantics at all, but rather use collocation-based statistical techniques (Yuret, 2004) or machine learning techniques (Decadt et al., 2004; Mihalcea and Csomai, 2005). Furthermore, SSI is an unsupervised approach, which makes it easy to add new terms as neologisms and jargon for disambiguation (i.e., there is no need of training). Moreover, in recent years, the SSI algorithm has turned out to be a promising and performing word sense disambiguation technique, as it performs better than other state-of-the-art unsupervised WSD methods in the Senseval-3 all-words and the Semeval-2007 coarse-grained all-words competition (Navigli, 2009).

Semantic similarity evaluation can be performed on numerous ways using distance measures (Jiang and Conrath, 1997; Lin, 1998; Maguitman et al., 2005; Resnik, 1995). Similar to Navigli and Velardi (2005), we make use of a simple, transparent, and intuitive distance measure which takes into account the length of paths between words in our semantic lexicon. The shorter a path between two arbitrary words in our semantic lexicon, the more similar we consider them to be.

The word sense disambiguation algorithm we propose in our current endeavors differs from the original SSI algorithm in a number of ways. First, we consider the two most likely senses for each word group and iteratively disambiguate the word group with the greatest weighted difference between the similarity of both senses to the context, rather than the word group with the greatest similarity for its best sense. Intuitively, this should yield better results than the original SSI, as it allows to consider the best separation of the senses of the to-be-disambiguated terms – picking the most similar sense might not be the best option if the similarity difference with respect to the next best sense is small. Furthermore, in case an arbitrary word cannot be disambiguated, we default to the first sense in our semantic lexicon (which in WordNet is statistically the most likely sense), whereas the original SSI algorithm fails to provide any word sense at all in such cases.

For an arbitrary news item, our algorithm (described in Algorithm 3.1) considers two lists of word groups. The first list d contains all word groups associated with only one sense, according to the semantic lexicon (WordNet), the ontology, and the already disambiguated word groups. The second list a contains all word groups with multiple possibilities for senses, i.e., the word groups to be disambiguated. The algorithm iteratively computes the similarity l of senses c of word groups in the second list to the senses s of word groups in the first list. The higher the similarity of a sense to already disambiguated

Algorithm 3.1: Word Sense Disambiguation.

```

1  a, d, s, c, l =  $\emptyset$ ;
2  w = getWordGroups();
3  foreach g in w do
4      senses = getSenses(g);
5      if  $|senses| == 1$  then
6          add(d, g); add(s, senses);
7      else
8          add(a, g);
9          foreach sense in senses do
10             if sense not in c then
11                 add(c, sense);
12             end
13         end
14     end
15   end
16   foreach sense in c do
17       simToS = 0;
18       foreach knownSense in s do
19           simToS = simToS + 1/shorestPathLength(sense, knownSense);
20       end
21       add(l, simToS);
22   end
23   lastAddedSense =  $\emptyset$ ;
24   disambiguate = true;
25   while disambiguate and a  $\neq \emptyset$  do
26       bestPick, bestPickSense =  $\emptyset$ ;
27       bestPickConf =  $-\infty$ ;
28       foreach g in a do
29           bestSense1, bestSense2 =  $\emptyset$ ;
30           bestSim1, bestSim2 =  $-\infty$ ;
31           senses = getSenses(g);
32           foreach sense in senses do
33               indexSense = indexOf(c, sense);
34               simToS = get(l, indexSense);
35               simToS = simToS + 1/shorestPathLength(sense, lastAddedSense);
36               set(l, indexSense, simToS);
37               if simToS > bestSim2 then
38                   if simToS > bestSim1 then
39                       bestSense2 = bestSense1; bestSense1 = sense;
40                       bestSim2 = bestSim1; bestSim1 = simToS;
41                   else
42                       bestSense2 = sense;
43                       bestSim2 = simToS;
44                   end
45               end
46           end
47           confidence = ((bestSim1-bestSim2)  $\times$  bestSim1);
48           if confidence > bestPickConf then
49               bestPick = g;
50               bestPickSense = bestSense1;
51               bestPickConf = confidence;
52           end
53       end
54       if bestPickConf > 0 then
55           rem(a, indexOf(a, bestPick)); add(d, bestPick); add(s, bestPickSense);
56           lastAddedSense = bestPickSense;
57       else
58           disambiguate = false;
59       end
60   end
61   foreach g in a do
62       rem(a, indexOf(a, g)); add(d, g); add(s, getSense(g, 1));
63   end

```

senses, the more likely this sense is assumed to be correct. The algorithm is initialized in lines 1 through 24. Then, each iteration, each word group in a is assessed by updating the similarity of its senses to s (lines 33 through 36) and identifying its best and second best senses (lines 37 through 45). Additionally, the word group with the greatest difference between the similarity of the best and second best sense (i.e., with the highest confidence) – weighted with respect to the similarity of the best sense – is identified (lines 47 through 52). When all word groups in a have been assessed, the best pick thus identified is disambiguated by taking the sense with the highest similarity to all disambiguated senses and moving the word group to the list of disambiguated word groups (lines 55 and 56), provided that this similarity is a positive number. In all other cases, the disambiguation process is terminated. If some ambiguous words remain by the time the disambiguation process finishes, our algorithm defaults to selecting their first WordNet sense (lines 61 through 63).

The similarity of a sense to already disambiguated senses is computed as the sum of the inverse of the shortest path length between this sense and the disambiguated senses in the WordNet graph. In our labeled directed graph representation of all senses of the considered word groups, we determine the shortest path between two concepts in a way which is similar to Prim's algorithm (Prim, 1957) for finding a minimum spanning tree for a connected weighted graph, an algorithm on which Dijkstra's algorithm (Dijkstra, 1959) is also based. Instead of computing a minimum spanning tree for the entire WordNet graph of the source and target concept, we compute two smaller spanning trees, having the source concept and the target concept as their root. We do this for both collections by iteratively walking to all direct neighbors of concepts considered in the collection, until a concept encountered in a walk in one collection has previously been encountered in the other collection.

In our running example, the *Word Sense Disambiguation* component adds the determination of noun and verb senses to the tokens' feature sets subsequently. These features contain numbers referring to the corresponding WordNet senses. Hence, no new annotations are added.

3.3.9 Event Phrase Gazetteer

When the meaning of word groups has been disambiguated, the text can be interpreted using semantics introduced by linking word groups to an ontology, thus capturing their essence in a meaningful and machine-understandable way. As we are interested in specific financial events, the *Event Phrase Gazetteer* scans the text for those events. It uses a

SAN FRANCISCO (Reuters) - Web search leader Google Inc. on Monday **said** it agreed to **acquire** top video entertainment site YouTube Inc. for \$1.65 billion in stock, putting a lofty new value on consumer-generated media sites.

Figure 3.8: *Event Phrase Gazetteer* annotations (phrases).

list of phrases or concepts that are likely to represent some part of a relevant event. For example, when we are looking for stock splits, we can search for ‘*stock split*’. Since the *Word Group Look-Up* component has already combined ‘*stock*’ and ‘*split*’ and the *Word Sense Disambiguator* has already assigned a concept value to this group of words, we can easily match this concept with events in our ontology.

The *Event Phrase Gazetteer* has some similarities with the *Ontology Gazetteer* since both of them try to find data from an ontology in a news message. In contrast to the *Ontology Gazetteer*, the *Event Phrase Gazetteer* takes annotated texts as input. Furthermore, the *Event Phrase Gazetteer* does not process the text lexically, but it looks for concepts, using the sense numbers that are assigned to the words in the text.

The look-up process takes place in two stages. First, the gazetteer is initialized by extracting all events from the ontology and linking them to the proper WordNet senses. This mapping is made accessible through a hash map, where a word sense can be used as a key to retrieve a reference to an event defined in the ontology. Second, at run time, the gazetteer iterates over the words in the text and uses the sense key (if any) to test whether a mapping to a corresponding event exists.

When processing our running example through the *Event Phrase Gazetteer*, we obtain the highlighted annotations – representing key concepts for possible events – shown in Figure 3.8. Since there are multiple types of events, in the features of these annotations a specification is given. Both the type of event is added, as well as the URI that points to the specific event in the ontology.

3.3.10 Event Pattern Recognition

Events thus identified by the *Event Phrase Gazetteer* are supplied with available additional information by the *Event Pattern Recognition* component, which checks whether identified events match certain lexico-semantic patterns (which are then used for extracting additional information related to discovered events). For instance, in case of a stock split, a concept indicating a company should precede the stock split keyword, and either before or just after the stock split keyword, a split-rate concept should be mentioned.

The *Event Pattern Recognition* component is based on the GATE *Rule Transducer* component, which uses JAPE (Cunningham et al., 2000) for manually defining patterns. JAPE provides a layer between the user and the regular expressions that are used internally. A typical JAPE rule consists of a pattern that has to be matched, followed by the commands that will be executed when that pattern is matched. These commands most of the time are comprised of a simple annotation command, but more powerful Java code is allowed too in the right hand side of the rule.

The following example of a JAPE rule extracts the proportions associated with a stock split event, e.g., ‘3-for-1’ (three new shares for one old share):

```

1 Rule: Props
2 (
3   ({Token.category == CD}) :new
4   ({Token.string == "-"})?
5   ({Token.string == "for"})?
6   ({Token.string == "-"})?
7   ({Token.category == CD}) :old
8 )
9 :prop --> :prop.Prop = {rule = "Props",
10                           new = :new.Token.root,
11                           old = :old.Token.root}

```

Line 1 specifies the pattern name, and lines 2 through 8 define the pattern to be searched for in the text. This pattern should consist of a cardinal number token (representing the number of new shares), followed by an optional ‘-’ token, a ‘*for*’ token, another optional ‘-’ token, and a cardinal number token (representing the number of old shares). Lines 9 through 11 specify the commands to be executed when the pattern is matched. The results from the pattern (i.e., the number of new shares, `new`, and the number of old shares, `old`) are stored into an annotation property.

By default, the GATE *Rule Transducer* only allows for simultaneous execution of JAPE rule files. If layering of rules (i.e., using one rule’s output as another rule’s input) is desired, an extra transducer has to be employed. In our implementation, we tackle this problem by feeding a JAPE rule file to the transducer that is nothing but a table of contents containing an ordered list of the different rule files that have to be executed. In this way, layering is possible, without being obliged to have multiple transducers in the pipeline. In addition to this, it enables easy recycling of useful blocks of rules.

In our running example, the *Ontology Gazetteer* already identified a subject and an object, namely ‘Google Inc.’ and ‘YouTube Inc.’, but those are not the subject and object of the sentence in a linguistic sense. To find the linguistic subject and object, the company

SAN FRANCISCO (Reuters) - Web search leader Google Inc. on Monday said it agreed to acquire top video entertainment site YouTube Inc. for \$1.65 billion in stock, putting a lofty new value on consumer-generated media sites.

Figure 3.9: *Event Pattern Recognition* annotations (subject, predicate, and object).

names are merged with the surrounding nouns, adjectives, and determiners. This is also done for verbs. For instance, ‘acquire’ indicates a buy event, but in order to have a better understanding of the sentence, the *Event Pattern Recognition* component annotates the predicate of the sentence by merging the **VerbEvent** annotation with the surrounding verbs, resulting in the annotations depicted in Figure 3.9.

Subsequently, after merging subjects, objects, and predicates, JAPE rules are matched to the annotated text. Whenever there is a match, the event pattern is executed, resulting in event annotations, e.g., **BuyEvent**, **DeclarationEvent**, etc. The annotation holds URIs to all important features of this event, including event type, event actors, and time stamp (derived from the news message). Figure 3.10 shows the final event annotation.

3.3.11 Ontology Instantiator

Finally, the knowledge base can be updated by inserting the identified events and their extracted associated information into the ontology using the *Ontology Instantiator*. At this phase, event instances are fully annotated in the text, which implies that no additional corrections need to be made. The module first retrieves a reference to the ontology by using the Jena (The Apache Software Foundation, 2013) library and then iterates over the available event annotations. Each time an event annotation is processed, an event instance is created in the ontology which belongs to a specific event class. Annotation features that are available are stored as properties of the individual. Furthermore, (relations between) concepts affected by the event are updated in the ontology. When the plug-in finished

SAN FRANCISCO (Reuters) - Web search leader Google Inc. on Monday said it agreed to acquire top video entertainment site YouTube Inc. for \$1.65 billion in stock, putting a lofty new value on consumer-generated media sites.

Figure 3.10: *Event Pattern Recognition* annotations (events).

execution, the ontology is again updated as it is enriched with new events originating from the processed text.

In the running example that is used throughout this section, we do not have to deal with a buy event, as an upcoming acquisition has only been announced. Therefore, a **DeclarationEvent** individual with its associated properties is created within the ontology. The relations between **Google** and **YouTube** can remain unchanged within the ontology. However, some of their properties are updated so that the ontology reflects Google's upcoming acquisition of YouTube.

3.4 Evaluation

In order to evaluate the performance of the implementation, we assess the quality of the individual pipeline components, each of which contributes to the output of the pipeline – i.e., annotations and events – and the pipeline as a whole. We measure the performance by means of statistics that describe, where applicable, latency and the cumulative error in terms of precision and recall. We define precision as the part of the identified elements (e.g., word senses or events) that have been identified correctly, and recall represents the number of identified elements as a fraction of the number of elements that should have been identified. When we compare the performance of different approaches, we assess the statistical relevance of differences in performance by means of a paired, two-sided Wilcoxon signed-rank test (Gibbons, 1986; Hollander and Wolfe, 2000), which is a non-parametric test evaluating the null hypothesis that the differences between paired observations are symmetrically distributed around a median equal to 0. If this null hypothesis is rejected, the compared samples are significantly different. This test would be suitable in this experimental setup, as the distribution of the values to be compared is unknown.

In our evaluation, we mainly focus on a data set consisting of 200 news messages extracted from the Yahoo! Business and Technology newsfeeds. In order to arrive at a golden standard, we have let three domain experts manually annotate the financial events and relations that we take into account in our evaluation, while ensuring an inter-annotator agreement of at least 66% (i.e., at least two out of three annotators agree). We distinguish between ten different financial events, i.e., announcements regarding CEOs, presidents, products, competitors, partners, subsidiaries, share values, revenues, profits, and losses. Our data set contains 60 CEO and 22 president discoveries, 232 statements linking companies with their products, partners, and subsidiaries, i.e., 136, 50, and 46, respectively, and 127 announcements of share values (45), revenues (22), profits (33), and losses (27).

Some components in our pipeline are existing, well-tested components, the performance of which has already been demonstrated in an extensive body of literature. However, one of the contributions of our current endeavors is that we propose several novel components that require a more detailed evaluation in terms of performance. The first component we evaluate in this respect is our *Ontology Gazetteer* component with our *OntoLookup* method, the performance of which we compare to the performance of the default GATE *OntoGazetteer* it replaces. The goal of both components is to identify lexical representations of concepts defined in an ontology. Precision and recall are not particularly useful here, as exact lexical representations known *a priori* (as is the case here) can always be identified in our corpus. Conversely, the latency is a more important issue in this component. On average, the *OntoGazetteer* needs 1.137 milliseconds (with a standard deviation of 0.265 milliseconds) per document to identify ontology concepts, whereas our *OntoLookup* method completes the same task in approximately 0.213 milliseconds (with a standard deviation of 0.039 milliseconds) per document. This significant 81% decrease (Wilcoxon *p*-value equals 0.000) in execution time needed can be attributed to the employed hash map trees.

Another newly proposed component utilizing hash map trees is our *Word Group Look-Up* component, which aims to identify compound words (i.e., word groups) in each document. If we do not use hash map trees in this component, but instead attempt to maximize our word groups by making numerous calls to our semantic lexicon in a linear search procedure, we need on average 68 milliseconds (with a standard deviation of 25 milliseconds) per document in our Yahoo! Business and Technology corpus for our task. Conversely, when we implement our proposed approach utilizing hash map trees, execution time needed decreases significantly with 46% (Wilcoxon *p*-value equals 0.000) to, on average, 37 milliseconds, with a standard deviation of 16 milliseconds.

Our *Word Sense Disambiguator* can be evaluated on a large, publicly available corpus designed specifically for this purpose – SemCor (Miller et al., 1994). We consider all 186 syntactically and semantically tagged SemCor documents containing 192,639 nouns, verbs, adjectives, and adverbs, which have been annotated with their associated POS, lemma, and WordNet sense. On this corpus, the original SSI word sense disambiguation algorithm exhibits an average precision of 53% with a standard deviation of 5 percentage points, a recall of 31% with a standard deviation of 9 percentage points, and an average execution time of 1,966 milliseconds, with a standard deviation of 755 milliseconds. Conversely, our proposed adaptation of SSI exhibits an average precision and recall of 59% with a standard deviation of 5 percentage points, as well as an average execution time of 2,050 milliseconds, with a standard deviation of 796 milliseconds. This implies an overall improvement in

Pipeline	Precision	Recall
SemNews (Java et al., 2006; Nirenburg et al., 2008)	68.00%	68.00%
ANNIE (Cunningham, 2002)	85.00%	85.00%
CAFETIERE (Black et al., 2005; Rinaldi et al., 2004)	84.03%	84.13%
KIM (Popov et al., 2004a)	86.00%	82.00%
SPEED	86.00%	81.00%

Table 3.3: Overview of the reported entity recognition precision and recall scores for several existing algorithms and information extraction pipelines.

precision with 12% and an improvement in recall with 90% in terms of the performance of the original SSI algorithm, while experiencing a mere 4% increase in execution time, which is just a matter of milliseconds. All observed differences are statistically significant, as they are all associated with a Wilcoxon p -value of 0.000, yielding a rejection of the null hypothesis of no difference between performance measures at a significance level of 0.001.

On our data set, our pipeline exhibits a latency of 632 milliseconds per document, with a standard deviation of 398 milliseconds. As for the output of the pipeline as a whole, we observe a precision for the concept identification in news items of 86% and a recall of 81%, which is comparable with existing systems. Table 3.3 shows the reported precision and recall for entity recognition for several existing information extraction tools, together with SPEED’s scores. Scores for other approaches are extracted from existing literature, as the individual tools are optimized for different purposes and therefore employ different data sets. As the evaluated data sets are different for each analyzed approach, the results presented in the table can merely be used as an indication of comparable performance, yet the table still underlines that in terms of precision and recall, SPEED’s performance is similar to existing (related) approaches. It should be noted that precision and recall of pipeline outputs, i.e., fully decorated events, result in lower values of approximately 62% and 53%, as they rely on multiple concepts that have to be identified correctly. To our knowledge, none of the existing approaches decorates identified events with their related information. As such, we cannot compare the final outputs of the considered approaches, as each approach in Table 3.3 has been designed for a distinct task.

Errors in concept identification result from missing lexical representations of the knowledge base concepts, and missing concepts in general. The disambiguator is supported by the *Word Group Look-Up* module, which identifies groups of nouns and verb phrases using WordNet. As a result of storing all data in a data base to keep it ready to use for future look-up, the more often the disambiguator is invoked, the faster the execution times will be (as concept similarities have been previously computed), thus eliminating a potential bottleneck. Despite using only WordNet as a semantic lexicon, we obtain high

precision as many of our concepts' lexical representations are named entities, which often are monosemous. High recall can be explained by SPEED's focus on detecting concepts from the ontology in the text, rather than on identifying all concepts in the text. The senses of word groups that are not present in the ontology are only used to help in the disambiguation of existing (already identified) concept lexical representations. The senses of the word groups not present in the ontology are not reflected in the precision and recall measures, as these measures only relate to identified ontological concepts (and their disambiguated senses).

3.5 Conclusions

We have proposed the Semantics-Based Pipeline for Economic Event Detection (SPEED), which aims to extract financial events from news articles (announced through RSS feeds) and to annotate these with meta-data, while maintaining a speed that is high enough to enable real-time use. For implementing the SPEED pipeline we have reused some of the ANNIE GATE components and developed new ones such as a high-performance gazetteer, word group look-up component, and word sense disambiguator. Although we focus on the financial domain, SPEED is generalizable to other domains, as we separate the domain-specific aspects from the domain-independent ones.

We have introduced a couple of novelties into the pipeline. Our pipeline components are semantically enabled, i.e., they make use of semantic lexicons and ontologies. Also, our WSD component employs a semantic lexicon (WordNet). Furthermore, the pipeline outputs results with semantics, which introduces a feedback loop; the knowledge base used within the pipeline can be updated when events are discovered, so that it represents the current state of the world. We thus incorporate learning behavior, making event identification more adaptive. Hence, the merit of our pipeline is in the use of semantics, enabling broader application interoperability. Other contributions lie within the speed of gazetteering and the improvements made to an existing word sense disambiguation algorithm (SSI). These novelties contribute to improved precision and recall.

However, since our framework is designed to deal with natural language, it may encounter noisy linguistic information. Our current framework is able to parse standard terms (which can be found in WordNet), as well as compound terms (which we identify by means of our novel word group look-up component). As future work, we aim to implement jargon terms by exploiting, e.g., Wikipedia redirects. Additionally, we plan to account for nonsense terms (i.e., misspellings) by using a similarity measure such as the Levenshtein distance. Alternatively, more extensive experiments regarding semantic sim-

ilarity evaluation (Jiang and Conrath, 1997; Lin, 1998; Maguitman et al., 2005; Resnik, 1995) are subject to future research, e.g., experiments with other similarity measures such as concept neighborhood, which is also applied in related domains (Tadesse et al., 2009), show promising results that could also be beneficial for our work.

Furthermore, a fruitful research direction would be related to the development of event trigger-based update languages (Lösch et al., 2009) for domain ontologies. Another suggestion for future research is to investigate event extraction rules learning from text using intelligent techniques (such as genetic algorithms). More interesting avenues for future work lie in investigating further possibilities for implementation in algorithmic trading environments (Allen and Karjalainen, 1999; Brock et al., 1992; Hellstrom and Holmstrom, 1999; Kearns and Ortiz, 2003). We aim to find a principal way of utilizing discovered events in this field. To this end, we also envision another addition, i.e., a way of linking sentiment (trends, moods, and opinions) to discovered events in order to assign more meaning to these events that can be exploited in an algorithmic trading setup. Sentiment of actors with respect to events may be the driving force behind their reactions to these events.

Chapter 4

Event Extraction Patterns[†]

THE Semantic Web aims to extend the World Wide Web with a layer of semantic information, so that it is understandable not only by humans, but also by computers. At its core, the Semantic Web consists of ontologies that describe the meaning of concepts in a certain domain or across domains. The domain ontologies are mostly created and maintained by domain experts using manual, time-intensive processes. In this chapter, we propose a rule-based method for learning ontology instances from text that helps domain experts with the ontology population process. In this method we define a lexico-semantic pattern language that, in addition to the lexical and syntactical information present in lexico-syntactic rules, also makes use of semantic information. We show that the lexico-semantic patterns are superior to lexico-syntactic patterns with respect to efficiency and effectiveness. Moreover, initial experiments show that learning patterns through a genetic programming approach results in higher quality rules than those resulting from a full manual approach within the same amount of time.

[†]This chapter is based on the article “W. IJntema, J. Sangers, F. Hogenboom, and F. Frasincar. A Lexico-Semantic Pattern Language for Learning Ontology Instances from Text. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 15(1):37–50, 2012.” and the conference publication “W. IJntema, F. Hogenboom, F. Frasincar, and D. Vandic. A Genetic Programming Approach for Learning Semantic Information Extraction Rules from News. In B. Benatallah, A. Bestavros, Y. Manolopoulos, A. Vakali, and Y. Zhang editors, *15th International Conference on Web Information System Engineering (WISE 2014), Part I*, volume 8786 of *Lecture Notes in Computer Science*, pages 418–432. Springer, 2014.”

4.1 Introduction

In today's information-driven world, many individuals try to keep up-to-date with the latest developments by reading news items on the Web. The contents of news items reflect past, current, and future world conditions, and thus news contains information valuable for various purposes. For example, being aware of current market situations is of paramount importance for investors and traders, who need to make informed decisions that could have a significant impact on certain aspects such as profits and market position. However, due to the ever increasing amount of information, it is virtually impossible to keep track of all emerging relevant news in an orderly fashion (Gross, 1964; Rampal, 1995). Hence, automatically filtering news items by means of computers would alleviate the cumbersome task of manually selecting relevant news messages and extracting information.

In contrast to human beings, machines (e.g., computers) are merely able to read news articles, not to understand them. With the Semantic Web (Berners-Lee et al., 2001), i.e., a collection of technologies that express and reason with content metadata, the World Wide Web Consortium (W3C) provides a framework to add a layer of semantic information to the Web, thereby offering means to help machines understand human-created data (e.g., news messages) on the Web. On the Semantic Web, metadata is defined using semantic information that is usually captured in ontologies, which are defined as shared formal specifications of conceptualizations (Gruber, 1993). Some of the most popular formats to describe ontologies on the Semantic Web are the Resource Description Framework (RDF) and RDF Schema (Brickley and Guha, 2004; Klyne and Carroll, 2004), and the Web Ontology Language (OWL) (Bechhofer et al., 2004). Ontologies can be used to store domain-specific knowledge in the form of concepts (i.e., classes or instances), together with associated inter-concept relations. These relations are denoted by triples that consist of a subject, a predicate, and an object.

Most of the current approaches to news filtering, such as the SeAN (Ardissono et al., 2001), YourNews (Ahn et al., 2007), and NewsDude (Billsus and Pazzani, 1999) frameworks, are able to retrieve only the news items that contain terms of the user's interest, not taking into account indirect information, which is also deemed relevant, such as competitors of companies of interest, political parties of politicians, etc. Exploiting the semantic contextual information related to concepts of interest enables a more comprehensive overview of relevant news with respect to certain topics. Therefore, in previous work (Frasincar et al., 2009; Schouten et al., 2010), we introduced the Hermes framework, which provides a method for personalizing news items that makes use of semantics. The framework stores lexicalized domain concepts and relations (i.e., properties that relate

concepts to each other or concepts to data types) in an ontology. Hence, Hermes stores synonyms or string representations of domain-specific entities (e.g., companies, persons, etc.) and their relations (e.g., subsidiary, competitor, etc.). The ontology is used for retrieving relevant news items in a semantically-enhanced way. In addition to this, we have proposed an ontology-based recommendation method that also benefits from a domain ontology (IJntema et al., 2010). As adding new information to an arbitrary but sufficiently large knowledge base requires a domain expert to invest a lot of time, in this chapter we propose a method that discovers new information automatically.

Automatic information discovery requires the use of information extraction techniques. In the last decades, a vast amount of research has already been conducted in this area. In general, information extraction can be done by means of statistics (Berger et al., 1996; Manning and Schütze, 1999; Taira and Soderland, 1999) or pattern-based rules (Hearst, 1992, 1998; Jacobs et al., 1991), each method having its own benefits and drawbacks. Statistical methods are mainly data-intensive, while pattern-based approaches usually are driven by knowledge more than data. From a user's point of view, large amounts of data are not always readily available, while (general) domain knowledge is usually at hand. As pattern-based approaches often require less training data than statistical methods, and also help users to gain more insight into why a certain relation was found, in this chapter we focus on pattern-based information extraction techniques. While the use of information extraction rules allows for semi-automatic information extraction, the construction of the patterns remains a non-trivial, tedious, and time-consuming process, because a trade-off needs to be made between the rules' precision and recall. Therefore, in this chapter, we additionally propose a method that assists the construction of information extraction rules.

In this chapter, we present a rule-based language that uses *lexico-semantic patterns* for information extraction. In contrast to *lexico-syntactic patterns* (Hearst, 1992, 1998; Hung et al., 2010), which combine lexical representations (i.e., strings) and syntactical information (e.g., parts-of-speech), lexico-semantic patterns also allow for the usage of semantic information such as concepts that are defined in ontologies. The notion of lexico-semantic patterns has already been introduced in our previous work. In (Frasincar et al., 2011a; Schouten et al., 2010) we extend the Hermes news processing framework by adding triple-based lexico-semantic event rules that make use of ontological concepts, in order to be able to recognize economic events. After validation, these events are subsequently coupled to the execution of action rules which update the underlying ontology. The use of lexico-semantic patterns for financial events discovery has also been discussed in (Borsje et al., 2010). There, we present a rule engine that allows for pattern creation based on

the triple paradigm (i.e., it makes use of a subject, a predicate, and an optional object), and that relies on triple conversion to the Java Annotations Pattern Engine (JAPE) language (Cunningham et al., 2000) and SPARQL (Prud'hommeaux and Seaborne, 2008). Last, in (Hogenboom et al., 2010d) we present a semantics-based information extraction pipeline for economic event detection, which makes use of lexico-semantic patterns that are defined in the JAPE language.

In the previous work discussed above, we consider mostly simple lexico-semantic patterns that are merely based on the triple paradigm, which hence makes it impossible to express more complex constructions. In this chapter, we present a more expressive language for specifying lexico-semantic patterns that makes use of regular expressions over ontology concepts. Furthermore, in our current endeavors, we aim for a simple, easy to use language for pattern creators. Existing languages like JAPE could easily result in verbose rules, while we aim for more compact ones. In addition, we give the formal specifications of our language and explain its constructs by means of examples, and we give a more extensive evaluation of the proposed pattern language in which we analyze the recognition of different types of events in textual representations. Last, we explore the application of an automated genetic programming approach for rule learning, in contrast to the manual processes of pattern creation elaborated on in our previous work.

By using lexico-semantic patterns that employ concepts and relations from a domain ontology, we aim to solve problems caused by ambiguity and specificity that exist in current approaches that employ lexico-syntactic patterns. The design of the lexico-semantic pattern language aims to fulfill the following requirements. First, the language should be developed for a Semantic Web context, where instances and their relations need to be learned from text. Then, the language should be accessible and easy to understand, yet expressive enough to be able to cover the required information extraction needs. By employing Semantic Web technologies, our language should remove some of the ambiguities inherent to lexical approaches, increasing the specifications precision level. In addition, a semantic approach allows to easily specify patterns that have many instances, increasing the recall of the information extraction process.

In this work, we aim to investigate the performance of lexico-semantic patterns compared to lexico-syntactic ones and for that, we evaluate the performance of both pattern languages by manually creating rules for each one of them that are subsequently applied for fact extraction from two distinct corpora consisting of news messages on financial topics and political topics, respectively. In this research, we consider facts to represent (financial or political) events like acquisitions, profit announcements, CEO changes, elections, provocations, etc., which are captured by triples consisting of a subject, a predicate,

and an object. We compare the performance of the languages with lexico-semantic patterns written in JAPE. Performance is measured in terms of construction times (i.e., efficiency) and in terms of precision, recall, and F_1 scores (i.e., effectivity). Last, we evaluate the performance of our rule learning approach on our financial corpus, measuring (improvements in) precision, recall, and F_1 scores.

The rest of this chapter is organized as follows. Section 4.2 discusses the related work, followed by Section 4.3, which elaborates on the Hermes Information Extraction Language (HIEL), i.e., the syntax for defining lexico-semantic patterns. Next, Section 4.4 describes our automated rule learning approach and Section 4.5 describes the Hermes Information Extraction Engine (HIEE). Our language and learning method are evaluated in Sections 4.6 and 4.7. Last, Section 4.8 concludes this chapter and identifies future work.

4.2 Related Work

In the current body of literature, various pattern grammars are described that could be of use in for instance news processing frameworks (Domingue and Motta, 2000; Java et al., 2006) or general purpose information extraction tools (Black et al., 2005; Cunningham, 2002; Manov et al., 2003; Popov et al., 2003, 2004b). These patterns are based on linguistic or lexical knowledge, as well as a priori human knowledge regarding the contents or topic of the text that is to be processed. We can make a rough distinction between two types of patterns that can be applied to natural language corpora, i.e., lexico-syntactic patterns and lexico-semantic patterns. The former patterns are a combination of lexical representations and syntactical information, whereas the latter patterns combine lexical representations with syntactic and semantic information.

4.2.1 Lexico-Syntactic Patterns

Hearst (1992, 1998) proposes the use of lexico-syntactic patterns for information extraction. This approach aims to find hyponym and hypernym relations by discovering regular expression patterns in free text. An example is the application of the following pattern to the sentence ‘... *works by such authors as Herrick, Goldsmith, and Shakespeare*’:

```
1 such NP as {NP,}* {(or|and)} NP
```

In this pattern, NP indicates a proper noun. Other text (i.e., **such**, **as**, **or**, and **and**) is used for lexical matching, while (and) contain conjunction and disjunction statements to

be evaluated, in this case a disjunction (denoted as `|`). Also, `*` is a repetition parameter that indicates the sequence between braces (`{` and `}`) is allowed to repeat zero to an infinite number of times. The rule presented above results in the following discovered relationships:

```

1  hyponym("author", "Herrick")
2  hyponym("author", "Goldsmith")
3  hyponym("author", "Shakespeare")

```

These patterns are often easy to comprehend by regular users, yet defining the right patterns to mine corpora to obtain unknown information is not a trivial task. Hearst stresses that, in order to return desired results successfully, patterns should be defined in such a way that they occur frequently and in many text genres. Also, they should often indicate the relation of interest and should be recognizable with little or no pre-encoded knowledge. Furthermore, all existing syntactic variations have to be included into a complex pattern to ensure its proper working.

4.2.2 Lexico-Semantic Patterns

Lexico-semantic patterns on the other hand are less cumbersome to define, as they make use of concepts instead of merely lexical representations, hereby alleviating the time-consuming process of pattern definition. In one of the first works introducing lexico-semantic patterns, Jacobs et al. (1991) propose a system that processes text prior to normal left-to-right syntactic parsing. The patterns may include terms and operators like lexical features, logical combinations, wildcards, and repetition, which are mostly adopted from the regular expression language. An example of a rule that will classify the verb phrase '*left dead*' as to express death or injury, is as follows:

```

1  ?PIVOT = (or found left shot)
2  ?OBJ    =* ?EFFECT=dead
3           => (mark-activator murder d-vp) ;

```

This sentence would also match '*found dead*' and '*shot dead*'. Next to standard elements such as repetition and wildcards, the rule presented here contains features like variable assignment on the left-hand side (LHS) (where words preceded by `?` denote variables) and on the right-hand side (RHS) macros such as `mark-activator`, which uses the results of the pattern match, including variable assignments, along with some other constants, such as `murder` and `d-vp`, to tag and segment the text. The main advantage of

such lexico-semantic patterns is that they take into account the domain semantics which help the parser cope with the complexity and flexibility of real text (Jacobs et al., 1991).

The lexico-semantic pattern language proposed by Jacobs et al. is similar to ours, since it also employs patterns for detecting semantics in text. Their framework is implemented in the GE NLToolset (Krupka et al., 1992), which is a set of text interpretation tools. In our framework we benefit from the natural language processing steps performed by GATE (Cunningham et al., 2002) and the underlying OWL ontologies. The software allows for easy extension and customization, in contrast to the GE NLToolset. In addition, we propose patterns that are easier to specify and comprehend by the end user than the patterns proposed by Jacobs et al..

In order to maintain readability, we aim for a notation similar to the one presented by Hearst. Even though our patterns add semantic functionalities, they strictly adhere to the standard POS tags (Robins, 1989) (in contrast to the patterns used by Jacobs et al.). Furthermore, our patterns require less keywords compared to the ones proposed by Jacobs et al., as they omit mark and pattern activators. Our intent is to explore the possibilities of adding semantics to the patterns by using Semantic Web technologies, and thus to make use of existing ontologies and support tools (e.g., reasoners, editors, readers, writers, etc.).

The Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RElations (CAFETIERE) framework as introduced by Black et al. (2005) is a rule-based system for ontology-driven text mining, which makes use of lexico-semantic patterns. CAFETIERE applies several preprocessing techniques to the text, i.e., tokenization, Part-Of-Speech (POS) tagging, and gazetteer lookup. To extract information from text, a rule notation is defined. A rule has the following form:

```
1 A => B \ C / D
```

where A represents the phrase that is recognized, B (optional) represents the text prior to C, C defines the text elements that are part of the phrase, and D (optional) is the neighboring text immediately following C. A basic example of a rule that would match an expression like ‘40 mg’ is:

```
1 [syn=NP, sem=QTY] => \[syn=CD], [sem=measure]/;
```

In this pattern, one is able to denote the characteristics of a matching token group, i.e., its syntactic category (i.e., a noun) and its semantic meaning (i.e., a quantity). In order to match an expression, the text should contain a token which is a cardinal digit, followed by a token that represents a measure. CAFETIERE also takes into account the

ordering of the rules. When one rule matches the text and annotates the text, the original annotation might no longer be visible to the next rule.

In our work, we benefit from the research that has been done in the context of the CAFETIERE project, e.g., by reusing parts of the developed rule notation. A limitation within the CAFETIERE framework is that rules are defined on a specific lexico-semantic level, i.e., semantic concepts are derived from an ontology (knowledge base) described in Narrative Knowledge Representation Language (NKRL) (Zarri, 1997). NKRL is a knowledge representation language which has been defined before the Semantic Web era, and has no formal semantics. Hence, the approach fails to properly describe domain semantics. Both the gazetteer and the lexico-semantic rules could benefit from an ontology-based approach, abstracting from the low-level and sometimes ambiguous lexical representations.

Another rule-based information extraction language that includes domain semantics is WHISK (Soderland, 1999). This language is based on regular expressions and can be used for extracting information from semi-structured text as well as free text. An example of a rule that extracts the number of bedrooms and the associated price for a rental ad is written as such:

```

1 Pattern:: * ( Digit ) 'BR' * '$'
2                               ( Number )
3 Output:: Rental
4           {Bedrooms $1}
5           {Price $2}

```

Whenever the pattern of the extraction rule matches a sentence, the syntactical elements that are enclosed by round brackets are being used as variables in the output statement. The first element `Digit` is assigned to `$1` and the second element `Number` is assigned to `$2`. In WHISK rules, the `*` symbol represents a wildcard, i.e., it is used to indicate an arbitrary sequence of characters without limitations to size and contents until the occurrence of the subsequent term in the pattern.

Even though WHISK does properly include domain semantics, the applicability of the language is limited. The support for wildcards creates flexibility in the patterns to be matched, but it is fairly restrained compared to for instance regular expressions. It is not possible to state a specific range of characters or words. Differently than WHISK, our language contains additional repetition operators, so that more expressive extraction rules can be created.

In (Maynard et al., 2002, 2007; Saggion et al., 2007) a Multi-Source Entity recognition system (MUSE) is proposed. This system employs the General Architecture for Text Engineering (GATE) (Cunningham et al., 2002) software, which is a Java-based environ-

ment supporting the research and development of language processing software, in order to extract information from text. The main focus is on the extraction of information from multiple sources and retain a certain robustness. The system consists of a number of components, including a tokenizer, gazetteer, sentence splitter, POS tagger, semantic tagger, and an orthographical matcher. The semantic tagging comprises a set of grammar rules based on the Java Annotations Pattern Engine (JAPE) language (Cunningham et al., 2000). An example of such a rule is:

```

1 Rule: GazLocation
2 (
3     {Lookup.majorType == location}
4 )
5 :loc --> :loc.Location = {kind = "unknown",
6                               rule = "GazLocation"}

```

In general, the LHS contains the pattern to be matched, whereas the RHS defines the action that is to be executed once a match has been found. This rule is fired (executed) when the gazetteer lookup results in a location. If this is the case, the pattern will be annotated with the type `Location` and two attributes, `kind` and `rule`.

Our work distinguishes itself from MUSE by proposing a language with a higher level of abstraction, which is easier to read for regular users. In addition to that, we focus on semantic patterns and aim to determine relations between concepts, rather than solely focusing on recognizing entities.

4.2.3 Pattern Learning

Since the composition of information extraction rules is a tedious process which requires a domain expert to invest a lot of time, a vast amount of effort has been put into automation of this process. We distinguish between supervised and unsupervised learning, where in the former method a model is learned from data of which the correct outcomes (classifications) are known, while the latter method does not rely on any prior knowledge. Due to the fact that on free text supervised methods generally perform better compared to unsupervised methods (Chang et al., 2006), we aim to employ a supervised learning technique. A problem many supervised approaches have to deal with, is the sparse amount of training examples, for which bootstrapping has proven to be an effective solution (Carlson et al., 2010).

Snow et al. (2004) learn hypernym relations from text using a supervised learning technique. The authors collect noun pairs from a corpus in order to identify new hypernym

pairs, and for each of these pairs sentences are gathered in which both nouns occur. New hypernym classifiers are trained based on patterns extracted from the gathered sentences, using classifiers like Naïve Bayes, genetic algorithms, and logistic regression. When such methods are applied in rule learning processes, rules are generated randomly during initialization and are altered in such a way that the built rules perform better in terms of a predefined metric, which is often a combination of precision and recall. With respect to rule generalization and specialization, Snow et al. distinguish between top-down and bottom-up approaches. The first type starts with a general rule and then aims to specialize it, while the latter starts with a specialized rule which is then generalized. Our approach goes beyond the one from Snow et al. by allowing domain concepts and relationships to be extracted from text.

WHISK (Soderland, 1999) employs a supervised top-down rule learning method. The rules learned in this system have been discussed earlier, and are based on regular expressions, which is similar to our approach. In addition to simple literals, syntactic and semantic tags are used to generalize the rules. In the learning process, these tags are determined by means of heuristics. While classes are allowed, no is-a hierarchy or other relationships are employed, while at the heart of our system is a domain ontology with both concepts and relations, which is used to create generic lexico-semantic patterns.

KNOWITALL (Etzioni et al., 2005) uses an unsupervised bottom-up approach to extract named-entities from the Web. The system employs patterns that incorporate POS tags to extract new information. Pattern learning is based on Web searches, where for each occurrence of an instance, a prefix of a specific amount of words and a suffix of a number of words is added to the pattern. The learned patterns consist only of an entity surrounded by words, unlike our approach, which employs a larger amount of linguistic information like orthographical categories and ontology elements, and not only POS tags. Furthermore, the expressiveness of the learned patterns appears to be limited, since repetition and logical operators are not allowed. KNOWITALL focuses on learning named entity extraction patterns rather than on the extraction of new relationships between entities, which is something we pursue.

While many of the above methods have proven to be effective when using lexico-syntactic rules, Genetic Algorithms (GA) are suitable for rule learning as well, since the input is often a bit string (Holland, 1992). One can encode a pattern such that every bit represents a token or its corresponding features. By employing different genetic operators, such as inheritance, selection, mutation, and cross-over, the optimal information extraction rule can be determined. A similar method is applied by Castellanos et al. (2010), who learn lexico-syntactic patterns that only incorporate POS tags, in order to extract

entities. This is inherently different from our approach, since we aim to generate lexico-semantic patterns to extract concepts, relationships, and events (complex concepts) from text.

A branch of Genetic Algorithms is Genetic Programming (GP), where generally each problem is represented as a tree instead of a bit string. This makes it easier to encode the problem. Each node either represents a sequence, a logical operator – e.g., conjunction, disjunction, and negation – or a repetition operator. Similarly, terminal tree nodes are suitable to represent a literal, syntactic category, orthographical category, or a concept. Genetic algorithms often converge fast to a good solution when compared to other meta-heuristics, such as simulated annealing (Thompson and Bilbro, 2000). By performing the default genetic operators, trees can evolve until the desired performance is achieved. In a similar manner, Borg et al. (2010) employ trees to represent rules, containing POS tags, that are used in genetic programming to discriminate between definitions and non-definitions in text. Because of the identified advantages of Genetic Programming approaches over other approaches, in our research, we use a Genetic Programming approach to pattern learning.

4.3 Hermes Information Extraction Language

The Hermes Information Extraction Language (HIEL) employs semantic concepts from an ontology. The language is evaluated in the context of extracting events and relations from news, as an extension to the existing Hermes news personalization framework (Frasincar et al., 2009; Schouten et al., 2010). This section continues by briefly explaining the characteristics of the Hermes framework as well as the usage of ontologies within the framework in Section 4.3.1. Subsequently, Section 4.3.2 introduces HIEL for semi-automatic information extraction from news items, of which also the Backus Naur Form (BNF) grammar is given in Appendix 4.A. Last, Section 4.3.3 elaborates on the usage of ontology elements within our language.

4.3.1 Hermes

Hermes (Frasincar et al., 2009; Schouten et al., 2010) is a framework that can be used for building a personalized news service. The framework enables users to select concepts from a knowledge base. Whenever these concepts, which could be individuals like **Microsoft** or **Google**, or related concepts, such as competitors, appear in an arbitrary news item,

the news item is presented to the user. Hence, the user will only be presented news items that match the user's interest.

Concept selection is performed by means of user-defined patterns. Similarly to CAFETIERE, Hermes is based on GATE and employs lexico-semantic patterns. However, these patterns use information from an OWL ontology that contains a schema of concepts and relations of various nature, thus making use of a standard language supported by many reasoners. Knowledge is stored in a separate ontological database that contains individuals. Each time a news message is processed, the ontology might be updated with new facts, so that the knowledge base remains up-to-date (Schouten et al., 2010).

The current knowledge base of Hermes is maintained by a manual approach. The domain ontologies are developed by domain experts. The process of developing the ontology is an incremental middle-out approach (Frasincar et al., 2009). Since news events can change the state of the world, each time such a change happens, the knowledge base should be updated. Because updating the ontology manually is a cumbersome process, it is preferred to do this at least semi-automatically. Therefore, we propose an information extraction language that can extract new individuals of concepts and relations from news items.

4.3.2 Language Syntax

The patterns previously proposed by Hearst (1992, 1998) serve as an inspiration for HIEL, as these lexico-syntactic patterns are easily comprehensible. Furthermore, these patterns provide the user with valuable insights into the reasons behind the extraction of certain information. Therefore, we aim to propose a language that approaches this simplicity, i.e., a language with which one is able to make patterns that are intuitive and easy to understand, but which also addresses the required expressivity. In this regard, it should have at least the expressivity of regular expressions. Our language can be characterized by supporting syntactic features, orthographic features, concepts, relations between concepts, logical operators, repetition, and wildcards. In this section, we explore the syntax of the language.

Language Definition

Typically, in HIEL, each pattern is described by a left-hand side (LHS) and a right-hand side (RHS). Once the RHS has been matched in the text to be processed, it is annotated as described by the LHS of the pattern. The LHS describes a relation between a *subject* (*sub*) and an *object* (*obj*) by using a *predicate* (*pred*). For example, *isCompetitorOf* is

a relation between the concepts `Microsoft` and `Google`. We denote the LHS of a pattern as follows:

```
1 (sub, pred, obj) :- RHS ;
```

The RHS, which is always terminated using a semicolon (;), describes a pattern that has to be identified in text. We define a pattern as an ordered collection of tokens that are divided by spaces, which indicates the sequence in which the target tokens have to appear in text. The RHS of a HIEL pattern is not limited to one sentence, but is matched against the full news article text. In order to limit a rule to a sentence, one has to specifically define this constraint in the pattern.

Prefixes

As both the LHS and the RHS can make use of ontological concepts, individuals, and properties that can be defined in different ontologies (identifiable through their unique namespaces), rules could easily become overly complex. In order to simplify the rule syntax, we introduce a short-hand method for namespace references. For this, HIEL allows for the definition of prefixes prior to the definition of the LHS and the RHS. For instance, a prefix `kb` for the ontology `http://www.hermes.com/knowledgebase.owl#` is created as follows:

```
1 PREFIX kb:"http://www.hermes.com/knowledgebase.owl#"
```

Note that such prefixes are particularly useful for pattern applications further explained in Section 4.3.3.

Literals

As shown in Section 4.2, pattern grammars typically support literals, i.e., text strings. Literals can be written as a (compound) word surrounded by quotes, e.g., ‘*John F. Kennedy*’. In HIEL, tokens on the RHS of patterns can be of various types, amongst which literals. Whenever literals are used within patterns, the (compound) word between quotes has to match exactly with the text.

Lexical Category

Like many other lexico-syntactic and lexico-semantic pattern languages, our language supports a set of syntactic categories to describe the lexical category of the token, i.e., its

Category	Description
CC	Coordinating conjunction
CD	Cardinal number
IN	Preposition
JJ	Adjective
NN	Noun
NNP	Proper Noun
PP	Pronoun
RB	Adverb
UH	Interjection
VB	Verb, base form
VBZ	Verb, 3rd person singular present

Table 4.1: Common lexical categories.

part-of-speech. The most common lexical categories are shown in Table 4.1. In general, we distinguish between various verbs and nouns, prepositions, adjectives, coordinating conjunctions (e.g., ‘as well as’), cardinal numbers, and interjections (e.g., ‘well’ as in ‘well, that depends’).

Orthographic Category

In addition to the word lexical category, the language distinguishes four orthographic categories. Note that the field of orthography spans hyphenation, capitalization, word breaks, emphasis, and punctuation. We define orthography as describing (defining) the set of symbols used in tokens. More specifically, we focus on capitalization. The `upperInitial` category is used for tokens that start with an uppercase character. When referring to capitalized words, `allCaps` should be used. In addition, `lowerCase` indicates a token without uppercase characters. Finally, `mixedCaps` is used in words with varying capitalization. Orthographic categories can especially be useful when identifying names or abbreviations.

Labels

The subject, relation, and object described in the LHS need to be identified in the RHS in order to provide a link between text and a new extracted fact. This can be done using labels, which are represented as words preceded by a dollar sign (\$) and followed by a colon and an equality sign (:=), as well as a description of the attached token. Whenever the RHS matches with a sentence, the tokens with associated labels are filled in the LHS of the rule. An example of a rule that employs labels is:

```

1 PREFIX kb:"http://www.hermes.com/knowledgebase.owl#"
2 ($sub, kb:hasProduct, $obj) :-
3     $sub:='Google' 'launches'
4     $obj:=upperInitial ;

```

In the example, the prefix `kb` refers to the namespace of a knowledge base (ontology) in which the predicate `hasProduct` has been specified, i.e., `http://www.hermes.com/knowledgebase.owl#`. This rule can be employed for finding new products of Google, and matches text fragments where the literals '`Google`' and '`launches`' are superseded by a token with the `upperInitial` orthographic category. Subsequently, `Google` and the latter token are bound to the labels `sub` and `obj`, respectively.

Logical Operators

The language supports three of the most common types of logical operators as defined by Gamut (1991), i.e., and (`&`), or (`|`), and not (`!`). The disjunction and conjunction are used in combination with grouping parentheses in the RHS. An example of such a rule is:

```

1 PREFIX rdf:"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2 ($sub, rdf:typeOf, $obj) :-
3     $sub:=(NN & upperInitial)
4     $obj:=(NN | CD) ;

```

Here, the subject (labeled `sub`) is a noun with an upper initial, and the object (`obj`) is either a noun or a cardinal number. The prefix `rdf` points to the namespace of RDF, which – amongst others – contains the `typeOf` property. Hence, the matching subjects are assumed to have a `typeOf` relation with their respective objects.

The logical operator indicating negation can be used freely in the RHS of a rule, yet it is not allowed to negate a label or a wildcard (to be discussed later). An example of a rule employing negation is:

```

1 PREFIX rdf:"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2 ($sub, rdf:typeOf, $obj) :-
3     $sub:=(!NN)
4     $obj:=(!(NN | CD)) ;

```

In the latter example, the subject is a token that is not a noun and which is followed by a token (the object) that is not a noun or cardinal number.

Repetition

Another feature that is often used in many languages is repetition, which is employed as an indication that a certain pattern can be found a number of times. In HIEL, we distinguish between four types of repetition operators: zero or more (*), once or more (+), zero or once (?), and a range ({min[, [max]]}). The latter indicates that the foregoing pattern must occur at least `min` times and no more than `max` times. The comma and the maximum are optional. When a maximum has not been defined, the pattern must occur at least `min` times. Leaving out the comma as well indicates that the specified pattern must occur exactly `min` times. An example of a rule utilizing a range operator is:

```

1 PREFIX rdf:"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2 ($sub, rdf:typeOf, $obj) :-
3     $sub:=NNP (VB | NN){1,3}
4     $obj:=NNP ;

```

This rule matches a text segment consisting of a proper noun (i.e., the subject), which is followed by 1, 2, or 3 tokens that are verbs or nouns, and by another proper noun (i.e., the object). As in our previous examples, the matched subject is assumed to have a `typeOf` relation with the matched object.

Wildcards

The patterns defined in the RHS of rules can be very specific. The order of the tokens is fixed and no other words between the tokens are allowed. In order to enable some flexibility in patterns, we allow the user to employ wildcards. These wildcards can be used to state that any word (token) may be found in the text and are inspired by the wildcards of the database query language SQL. Within our language, a wildcard is denoted with an underscore (_), which can optionally be followed by repetition operators. As stated earlier, it is not allowed to precede the wildcard with a negation operator. An example rule that makes use of wildcards is:

```

1 PREFIX rdf:"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2 ($sub, rdf:typeOf, $obj) :-
3     $sub:=(NN & upperInitial) _{5}
4     $obj:=NN ;

```

In the latter example, the subject, i.e., a noun token starting with an uppercase character, is followed by exactly 5 wildcards and another noun which is bound to the object of the rule.

4.3.3 Employing Ontology Elements

By employing ontology elements, we are adding semantics to the rules. For instance, let us assume we are interested in discovering new occurrences of a `kb:Company` introducing a new `kb:Product`. If there is a news article about `kb:Google` introducing a new product, e.g., `kb:Chrome`, and `kb:Google` already has an entry in the knowledge base (ontology), it is possible to annotate the lexical representation of `kb:Chrome` as a product and add a product-relation between `kb:Chrome` and `kb:Google`. When ontologies are employed in the rules, potentially one rule can be used to describe multiple lexical representations. In this example three features of an ontology occur. First, `kb:Company` and `kb:Product` are *classes*. Second, `kb:Google` and the product (`kb:Chrome`) are *individuals* of these classes, and third, the relationship between `kb:Google` and the product represents an *object property*. We now continue by discussing how these three features of the ontology can be employed in information extraction rules.

Concepts

Classes are groups of individuals that share the same properties (Bechhofer et al., 2004). For example, `kb:Google` and `kb:Microsoft` both belong to the class `kb:Company`. Other examples of classes are `kb:Product`, `kb:Person`, and `kb:Country`. In information extraction it is useful to look for specific individuals, rather than classes, in text fragments. Individuals are more specific than classes and are generally used on the RHS of the rule.

In the language we make a distinction between classes and individuals. If the rule is to recognize a specific individual of a certain concept it is denoted by the individual itself. The following rule shows an example:

```

1 PREFIX kb:"http://www.hermes.com/knowledgebase.owl#"
2 ($sub, kb:hasProduct, $obj) :-
3   $sub:=kb:Google kb:Buys
4   $obj:=mixedCaps ;

```

This rule contains two individuals, namely `kb:Google` and `kb:Buys`, and are matched to a sentence like ‘*Google Inc. acquires YouTube*,’ because ‘*Google Inc.*’ is a lexical representation of the instance `kb:Google` and in a similar manner is ‘*acquires*’ a lexical representation of `kb:Buys`. By employing classes instead of specific individuals, the rules become more generic. An example of a rule using classes is:

```

1 PREFIX kb:"http://www.hermes.com/knowledgebase.owl#"
2 ($sub, rdf:typeOf , kb:Company) :- 
3     [kb:Company] (',' | 'and')
4     $sub:=(NNP{1,}) ;

```

In the latter rule, a list of companies is recognized. The square parentheses denote that any of the individuals of the enclosed type may be matched. Each individual has associated lexical representations as we have previously seen. In this example, the proper nouns (NNP) will be annotated as an individual of a company. Assuming that `kb:Google` is already known as a concept, in order to recognize other companies, we can match the rule on the sentence '*A Big-Picture Look at Google, Microsoft Corporation, Apple and Yahoo!*'. The first time this is done, '*Microsoft Corporation*' will be annotated as a company, while in order to recognize '*Apple*' and '*Yahoo!*' as well, the rule needs to be run a second and a third time.

Relations between Concepts

As stated earlier, the LHS of the HIEL patterns is used for recognizing concepts, and it is a triple that describes the relationship between a subject and an object. By using labels, we can refer in the LHS to a concept found on the RHS. For instance, a rule such as

```

1 PREFIX kb:"http://www.hermes.com/knowledgebase.owl#"
2 ($sub, kb:hasSubsidiary , $obj) :- 
3     $sub:=[kb:Company] kb:Buys
4     $obj:=[kb:Company] ;

```

can be employed in order to extract the `kb:hasSubsidiary` relation between two companies. The individual `kb:Buys` has various synonyms such as: '*buy*', '*acquire*', and '*take over*'. If we apply this rule to the sentence '*Google buys YouTube for \$1.65 billion*', it would extract the `kb:hasSubsidiary` relation between `kb:Google` and `kb:YouTube`, assuming that the corresponding tokens have been annotated with their correct ontology individuals `kb:Google` and `kb:YouTube`, respectively, which belong to the class `kb:Company`. This information can then be used in order to update the ontology, for instance for removing the existing competitor relationship between the companies.

4.4 Rule Learning

In order to assist domain experts with rule creation, we propose to employ a genetic programming approach to rule learning. Our information extraction language, HIEL, which

can intuitively be implemented using tree structures, fits the required tree structure of the genetic programming operators. Additionally, a genetic programming approach offers transparency in the sense that it gives the user insight into how information extraction rules are learned. Also, a genetic approach – as opposed to other meta-heuristics such as simulated annealing – often converges to a good solution in a relatively small amount of time (Thompson and Bilbro, 2000).

4.4.1 Rule Learning Process

Figure 4.1 depicts the basic steps of our genetic programming approach to rule learning, where each circle represents a rule. First rules are initialized, followed by the evaluation of the fitness of each of these rules. Rule evolution is done by applying a genetic operator on the rules, i.e., elitist selection, cross-over, and mutation. Based on a selection procedure which takes into account the fitness of individuals we determine the rules on which these operators are applied. This process continues until one of the termination criteria is fulfilled, after which the rule with the highest fitness is collected in a rule group. The

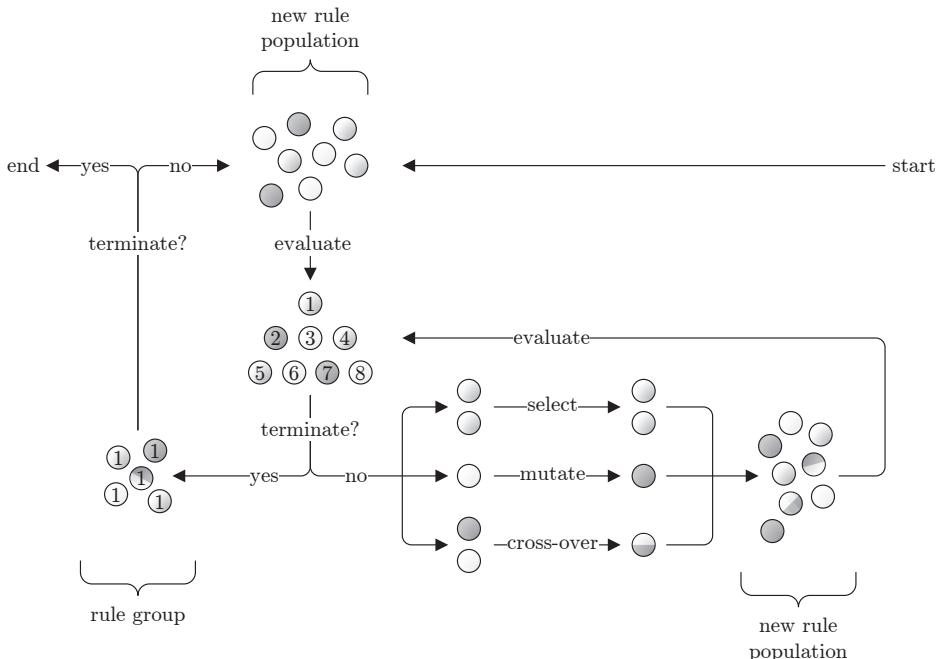


Figure 4.1: Rule learning process.

latter is a group of rules, in which each rule aims to extract the same type of information, albeit covering different situations. Since a single rule is not likely to achieve a high recall, because of the many different sentence structures, a collection of rules could achieve this goal. After the best rule, i.e., the rule with the highest fitness, has been selected and the rule group does not yet meet its termination requirements, a new population is initialized and another iteration is performed.

4.4.2 Representation

While in genetic algorithms, individuals are generally encoded in the form of an array of bits, in genetic programming individuals are specified as trees. In our representation, a tree consists of two types of nodes, i.e., functions and terminals. The first type has functions and terminals as children, whereas the second type cannot have child nodes. We differentiate between five functions, i.e., a sequence, a conjunction operator, a disjunction operator, a negation operator, and a repetition. Also, we distinguish four terminals, i.e., a syntactic category, an orthographic category, a concept, and a wildcard.

Each information extraction rule can be represented by a tree. Considering the fact that HIEL requires labels to be placed on separate elements on the first level of the tree and each label should be bound to different tokens in the text, the root of each tree is a sequence, which can have one or more child nodes of type function or terminal.

4.4.3 Initialization

The first phase in the genetic programming process is the initialization of the rules. During the initialization, a population of N individuals is created. For initialization, each node needs to be created such that it is syntactically correct. In addition, a maximum number of nodes per tree and a maximum tree depth helps constraining the rule size and complexity.

Generally, in genetic programming, information extraction rules are generated randomly at initialization phase. A commonly used method is ramped-half-and-half, which is proven to produce a wide variety of trees of various sizes and shapes. The ramped-half-and-half initialization procedure consists of two methods, i.e., *full* and *grow*. The full method generates trees for which the leaves (terminal nodes) are all at the same level (i.e., maxdepth), while the grow method generates more variously shaped trees. Because neither of the methods provide a wide variety of individuals, half of the population is constructed using the full method and half of the population is constructed using the grow method.

4.4.4 Fitness Evaluation

Each individual in the population is evaluated for each generation in order to determine its fitness. We compare the extracted information with manually annotated information by evaluating the F_1 -measure and the number of nodes within a tree. The F_1 -measure is defined as the harmonic mean of precision (correctly found items) and recall (correctly found items with respect to should-be-found items). We calculate the number of nodes within a tree in order to control the amount of bloat (i.e., uncontrolled growth of information extraction rules during the evolutionary process) in the population. Both measures are combined into one fitness measure that determines how well an individual performs compared to others.

A common problem in genetic programming is tree size explosion. Often, rules are learned that have the same fitness, but that are slightly different. In order to overcome the problem of learning rules consisting of unnecessary nodes, we introduce some parsimony pressure by including a small penalty in the overall fitness measure for the total number of nodes of the rule. Let α denote the amount of bloat and R represent a rule, then the fitness of a rule (when taking into account both F_1 and rule length l) is determined as:

$$\text{Fitness}(R) = \begin{cases} 0 & \text{if } F_1(R) = 0 \\ \frac{\alpha}{l(R)} + (1 - \alpha) \cdot F_1(R) & \text{if } F_1(R) > 0 . \end{cases} \quad (4.1)$$

4.4.5 Selection

For each genetic operator one or more individuals from the population need to be selected. According to the Darwinian principles, the strongest individuals survive, therefore it is better to select individuals based on their fitness. A common selection method is tournament selection. One of the advantages of this method is that the selection pressure, which determines the degree to which it favors fit individuals over less fit individuals, remains constant. In tournament selection, ts (tournament size) individuals are selected randomly from the population. These selected individuals are then compared with each other, and the one having the highest fitness wins and is selected. By adjusting the tournament size, the selection pressure can be adapted.

4.4.6 Genetic Operations

After the rules have been initialized, the actual process of evolving can be initiated. During the evolution, several genetic operators are applied, i.e., elitist selection, and reproduction through cross-over and mutation.

Elitist Selection

The first operation, elitist selection, resembles the survival of the fittest principle from Darwin. After the fitness of each individual in the population has been determined, the best r performing individuals are selected and copied to the next generation. The user may alter the portion of the population that is allocated for selection. Generally r is set to a value between 5% and 10%, in order for the algorithm to keep just a small set of the best performing individuals.

An advantage of applying the selection operator is that it helps the process to remember the best performing individuals until a better one is found. If the operator is omitted, these well performing rules might disappear from the population due to the reproductive cross-over and mutation operators.

Cross-over

During the cross-over operation two parents are selected from the population to produce either one or two offsprings. The former method randomly selects a cross-over point in both parents and interchanges the selected nodes, producing two children. The latter also randomly selects a cross-over point in both parents, but generates one child by combining the selected parts from both parents. Each parent is chosen based on its fitness using tournament selection and could be selected more than once in each generation, making it possible to use the same individual for multiple cross-over operations.

The selection of the cross-over points is generally not done with uniform probability, since the majority of the nodes will be terminal nodes. In order to overcome this problem, we select 90% of the time a function and 10% of the time a terminal node. While individuals are selected based on their fitness, the nodes interchanged during cross-over are selected in a random manner. This can result in offspring that do(es) not necessarily perform well, while the originating trees can have a relatively good performance. This is the case if a node (including its child nodes), also called a subpattern, is almost never discovered in the text.

Mutation

The mutation operator aims to introduce more variety into the population. Several approaches are identified in mutation for genetic programming. The first is subtree mutation, also known as headless chicken cross-over, where a random point in the tree is replaced by a randomly generated subtree. A second approach is point mutation. In this method only the randomly selected point is replaced by a function or terminal. If no replacement

is possible (i.e., if the randomly generated node is not allowed within the selected parent node), the mutation is not performed. We utilize the headless chicken cross-over method, because of its reported good performance with respect to the other approaches (Angeline, 1997; Jones, 1995).

4.4.7 Termination Criteria

A genetic programming run terminates when one of the termination criteria is satisfied. We distinguish between two termination criteria, i.e., one for a run and one for a rule group. Each run generates a maximum number of generations, which can be specified by the user. Because of the wide variety in sentence structures it is not plausible that one rule would be able to achieve high recall and precision values, yet a group of rules might be able to achieve this goal for a particular event. Once a termination criterion has been fulfilled, the rule with the highest fitness is saved to the assembled rule group, i.e., a set of rules that intend to extract the same information (i.e., triple type). For example, it is likely that one needs several rules to extract all instances of a `kb:hasCEO` relationship between a `kb:Company` and a `kb:Person`. At least two rules are needed to extract both the instance in '*Apple's chief executive, Steven P. Jobs*' and '*Steve Ballmer, Microsoft's chief executive*' as the order of the company and the CEO is different in these two cases.

Once a rule is learned and added to the rule group, the information extracted by this rule is excluded while learning additional rules. If a rule does match a previous annotation, it is not taken into account for its fitness, and hence each rule will extract different information. After the termination criterion for the current population fires, the rule with the highest fitness is only collected in the rule group if it causes the rule group to achieve a higher overall fitness value. In case it lowers the fitness of the entire group, it is omitted. The entire rule learning process, i.e., assembling the rule group, terminates when T iterations of updates have passed in a sequence, which did not manage to produce rules that increased the fitness of the rule group, meaning the algorithm is stuck in a (local, possibly sub-optimal) solution.

4.5 Hermes Information Extraction Engine

Based on the language defined in this chapter, we have implemented the Hermes Information Extraction Engine (HIEE). In this section, we first discuss the Hermes News Portal (HNP) in Section 4.5.1, followed by Section 4.5.2 that briefly touches upon the general framework that lies underneath the HNP and the HIEE plug-in. Subsequently,

Section 4.5.3 presents the preprocessing of the news items. Section 4.5.4 discusses the rule engine and Section 4.5.5 illustrates the rule development and learning plug-in for the Hermes News Portal.

4.5.1 Hermes News Portal

The implementation of the Hermes framework is the Hermes News Portal (HNP), which allows users to formulate queries and execute them on the domain ontology in order to retrieve relevant news items. The HNP application is a stand-alone, Java-based tool which makes use of various Semantic Web technologies.

The internal knowledge base is in fact a domain ontology constructed by domain experts, represented in OWL (Bechhofer et al., 2004). While populated ontologies are typically queried by the Semantic Web’s standard query language SPARQL (Prud’hommeaux and Seaborne, 2008), querying within HNP is done by means of extended SPARQL queries. Because within the Hermes News Portal time-specific features are exploited, time functionalities were added to SPARQL, which resulted in tSPARQL (Frasincar et al., 2009; Schouten et al., 2010). Within HNP, the classification of the news articles is done using GATE (Cunningham et al., 2002) and the WordNet (Fellbaum, 1998) semantic lexicon. The classification occurs prior to the rules execution that extract information from news.

4.5.2 General Framework

We developed a general framework that supports our Hermes Information Extraction Engine (HIEE) plug-in. Figure 4.2 presents the architecture of the processing pipeline that lies underneath our implementation. The pipeline takes as inputs news documents (for instance originating from RSS feeds) and rules (either specified by the user or by the genetic programming algorithm), and is centered around an ontology. The framework further consists of two main parts, i.e., the preprocessing stage and the rule engine. These parts and their individual components are executed in a specific order, and are discussed in more detail in the following sections.

In short, preprocessing – which is described in more detail in Section 4.5.3 – is done using the existing HNP natural language processing pipeline, which classifies news items using the GATE architecture (Cunningham et al., 2002). Most of the components stem from the A Nearly-New Information Extraction (ANNIE) system, which is a selection of standard GATE components. In addition, an ontology-enabled gazetteer is employed.

In contrast to most preprocessing components, the rule engine, which is described in more detail in Section 4.5.4, makes use of ontologies. The engine consists of two core

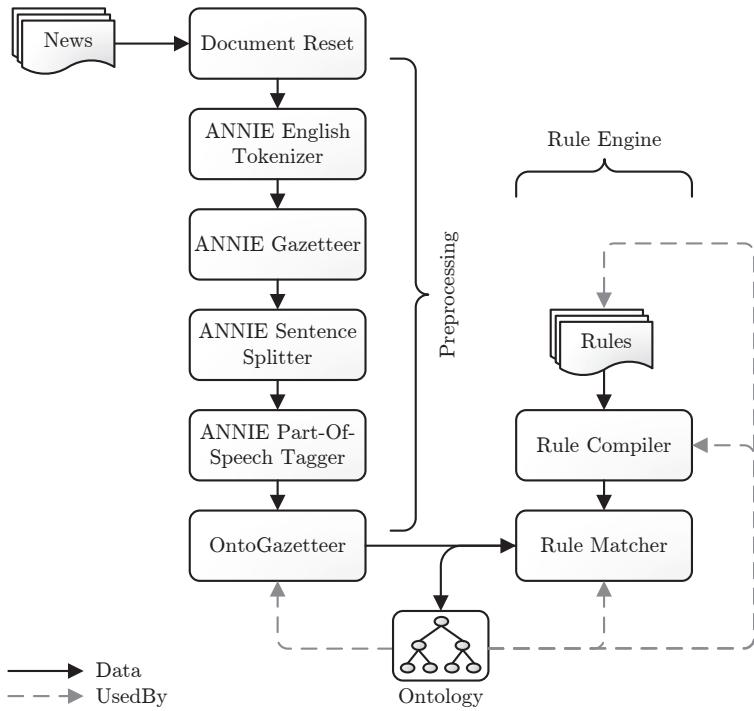


Figure 4.2: Overview of the Hermes processing pipeline.

components, i.e., lexico-semantic pattern (rule) compilation and matching. The compiler and matcher make use of semantic components, i.e., concepts and individuals stored in the main ontology, and syntactic elements, such as part-of-speech tags that are generated in the preprocessing stage.

4.5.3 Preprocessing

Before the rules can be employed to match patterns in text, a few processing tasks need to be performed, like tokenization, sentence splitting, and part-of-speech tagging, which are dealt with by the GATE architecture (Cunningham et al., 2002). GATE provides a pipeline consisting of different components, each of which handles a different aspect of the language processing. The components that are part of the pipeline, and come with GATE by default, are in order of usage: *Document Reset*, *ANNIE English Tokenizer*, *ANNIE Gazetteer*, *ANNIE Sentence Splitter*, *ANNIE Part-Of-Speech Tagger*, and *OntoGazetteer*.

The *Document Reset* component is used for resetting the document, in this case a news item, to its original state. The document is cleared from all its current annotations, enabling the pipeline to re-annotate the text. This is especially useful when running the document through a pipeline several times, as it is undesirable to use a document with previous annotations in an information extraction process. Subsequently, the *ANNIE English Tokenizer* splits the corpus into tokens, such as numbers, punctuation, and words of different types. A distinction is made between words in uppercase and lowercase, and between certain types of punctuation.

After these basic operations, the *ANNIE Gazetteer* looks up words from gazetteer lists (i.e., lists with names of, for example, cities, countries, companies, days of the week, world leaders, etc.) in order to be able to classify them. In our implementation, the latter task is limited to some basic and static lists, such as days of the week, months of the year, etc. After gazetteering, the *ANNIE Sentence Splitter* is employed, which identifies sentences, required for the *ANNIE Part-Of-Speech Tagger*. This tagger is a modified version of the Brill tagger (Brill, 1992), which produces a POS tag as an annotation to each word or symbol. The POS tags, e.g., the ones described in Table 4.1, can be used in the rules to describe certain patterns.

Finally, the *OntoGazetteer* component is executed, which has similarities with the ANNIE Gazetteer. The biggest difference lies in the fact that the *OntoGazetteer* is an ontology-enabled component, i.e., it utilizes terms stored in an ontology instead of plain gazetteer lists for classification. The component still utilizes lists in order to perform its tasks, but in addition provides a mapping definition between the lists and the ontology classes. The *OntoGazetteer* searches the corpus for occurrences of OWL annotation properties – these are the concept lexical representations – of the classes and instances of the ontology. Similar to our previous efforts (Hogenboom et al., 2013b) discussed in Chapter 3, the gazetteer operates on unprocessed tokens (and hence does not make use of POS tags, lemmas (not considered here), etc. Therefore, the position of the gazetteer is not important, as long as it is placed after the *ANNIE English Tokenizer*. The found matches are annotated with the name of the OWL individual (or class) against which the piece of text is matched. In order to assure a good performance, one should make sure that the ontology has an extensive list of lexical representations associated to each depicted concept or relation. After annotation using the *OntoGazetteer*, tokens have been linked to the ontology, and hence can be used in lexico-semantic patterns. A sentence like ‘*The conference will be attended by Microsoft and Apple CEOs Steve Ballmer and Steve Jobs*’, gives us the opportunity to recognize ‘Steve Ballmer’ and ‘Steve Jobs’ as CEOs of ‘Microsoft’ and ‘Apple’, respectively, e.g., as individuals `kb:SteveBallmer` and

`kb:SteveJobs` of class `kb:Person` and `kb:Microsoft` and `kb:Apple` of class `kb:Company` have been annotated and hence can be employed to extract relations of type [`kb:Company`] `kb:hasCEO` [`kb:Person`].

4.5.4 Rule Engine

After preprocessing a news corpus, the Hermes Information Extraction Rule Engine compiles the rules in the *Rule Compiler* and matches these rules to the text using the *Rule Matcher*. Because we use news items, employing the extracted information it is possible to adapt the underlying ontology based on certain events. For instance, ‘*Eric Schmidt leaves Google*’, informs us that ‘*Eric Schmidt*’ is no longer the CEO of ‘*Google*’ and hence results in an ontology update regarding `kb:EricSchmidt` and `kb:Google`. Note that in order for the rule engine to be able to run as a stand-alone application, we do not create dependencies with respect to GATE’s default JAPE language. Hence, because no conversion is made to JAPE rules, we enable one to employ the rule engine within other information extraction frameworks as a stand-alone component. Also, we take into consideration that JAPE might not be suitable to support possible future extensions to HIEL.

The *Rule Compiler* is created using the Java Compiler Compiler (Sun Microsystems, 2013), developed by Sun Microsystems. The Java Compiler Compiler generates a compiler for the grammar defined in Section 4.3 and Appendix 4.A. During the compilation, Java objects are being created that represent the various parts of a rule. The right-hand side (RHS) of a rule can be represented as a tree, as shown in Figure 4.3. Components in this tree are of two main types: internal nodes and leaf nodes. Internal nodes consist of one or more internal nodes or leaf nodes and include sequences, logical operators, and repetitions. Leaf nodes are nodes that do not have any child nodes and include literals, concepts, orthographical categories, part-of-speech categories, and wildcards. After the rules are compiled, the matcher tries to match the rules onto the text.

In order to match the compiled rules to the text, each tree node performs its own task. In our recursive algorithm which starts at the tree’s root node, child node procedure calls are performed. These children try to match as many tokens as possible. Non-leaf nodes, i.e., nodes that contain child nodes, keep performing calls to their children until a leaf node has been reached. Subsequently, leaf nodes check whether the token at the current position is a match. Each child node reports to its parent the number of tokens it was able to match until the root of the tree is reached. If the root returns a value which is equal or greater than the value of the position it started with, the rule has been matched to the text. This process is repeated until the last token of the text has been reached.

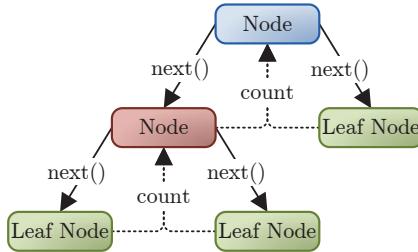


Figure 4.3: Rule tree template.

After matching a rule, tokens on the right-hand side are bound to labels to be used in the left-hand side of the rule. This allows for determining which tokens belong to the subject, predicate, and object of the computed triple.

In Figure 4.4 an example tree is shown that corresponds with the example rule presented in Section 4.3.3, extracting a list of companies. If we consider the following sentence: ‘ASUS and Microsoft Corporation become official partners for Windows Phone 7’, where `kb:ASUS` (with an associated lexical representation ‘ASUS’) is a known individual of `kb:Company` in the ontology, and ‘Microsoft Corporation’ does not match any lexical representation, the process is as follows. `Sequence` sends a `next()` call to `[kb:Company]`, which returns 1, indicating that one token has been matched. Subsequently, after receiving the response, the `Sequence` sends a `next()` call to the `OR` which passes it on to the literal ‘and’, which returns 1. Finally, the `Repetition` tries to match the `NNP` as many times as possible (with a minimum of 1), which results in 2. Note that the tokens matched by the `Repetition`, ‘Microsoft Corporation’, are assigned to the left-hand side (LHS) entity `sub`.

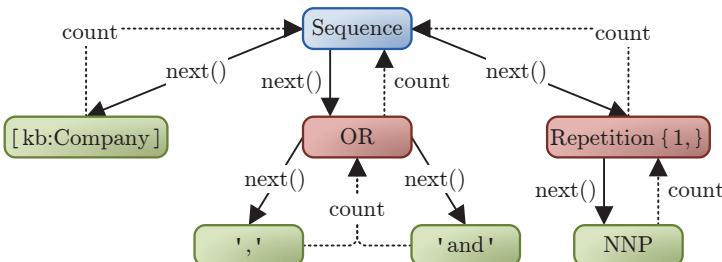


Figure 4.4: Rule tree example.

Regarding speed, the Hermes Information Extraction Rule Engine is able to run on a real-time basis, as both rule compilation and execution for one news message have subsecond performance. We did not encounter any speed issues that can be attributed to OWL operations on the underlying ontology, as we only deal simple inferences based on the `typeOf` relations.

4.5.5 Hermes Plug-in

In order to be able to evaluate the usability and expressivity of the proposed information extraction language and learning algorithm, the HIEE plug-in for the Hermes framework was created. This plug-in allows one to create, edit, learn, use, and evaluate extraction rules, and is composed of four different parts, i.e., a rule editor, an annotation validator, an annotation editor for manual annotations, and a rule learning environment.

The rule editor allows users to create their own personal information extraction rules. These rules can be divided into so-called rule groups, enabling clustering of different rules of the same type (i.e., they discover the same type of event). After creating a rule, the user is given the option to validate and save the rule. Whenever syntactical mistakes – such as typographical errors, but in worse cases violations of the grammar as defined in this chapter – are made by the user, the built-in compiler will detect them and display informative messages to the user. A rule cannot be saved if it is not valid, ensuring the validity of the rules by construction.

The annotation validator is developed to ensure semi-automatic ontology updates. It displays the resulting annotations after applying the user-defined extraction rules to the existing news items, together with their associated number of occurrences. After approval by the user, ontology updates are performed. In subsequent annotation runs, the newly extracted (and approved) facts are incorporated into the knowledge base, yielding more accurate and up-to-date results.

The annotation editor is used for evaluating the current rule set. For each news item, the user is able to manually annotate tokens from the news item. Events can be described by selecting the subject, predicate, and object in the text and by annotating them with the corresponding ontology concepts. Several aiding mechanisms are available, such as an overview of current classified annotations of a selected token, such as the part-of-speech tag, the orthographical category, and the ontology concepts.

The plug-in additionally features a rule learning environment, in which rules are created following the genetic programming approach introduced earlier in this chapter. The user is able to keep track of the current generation, the learned rules, and their fitness.

Several controls are put in place for managing the rule learning process. Additionally, current generations and learned rules are displayed. Last, the user is able to fine-tune the algorithm parameters.

4.6 Evaluation of Manually Created Patterns

In order to evaluate the effectiveness of our language, we have implemented a test method and built a test environment. First we discuss the evaluation setup in Section 4.6.1, followed by the results, in Section 4.6.2.

4.6.1 Evaluation Setup

For testing the performance of the extraction language, we assembled news messages from financial news feeds, totalling 500 items with an average length of 4,200 words and multiple paragraphs. News messages are written in English using an extensive vocabulary. These news items are divided into two sets, i.e., a training set consisting of 300 news items, and a test set consisting of 200 news items. The gathered news items originate from Reuters Business and Technology News and from The New York Times Business News. Next, an ontology is provided to domain experts (i.e., colleagues with an expertise in finance) that are asked to annotate the news messages and to develop event extraction rules. A similar approach is followed for a second data set containing 100 political news messages, with an average length of 700 words, mainly gathered from Reuters Politics News and Yahoo! Politics News.

The ontologies employed in our experiments contain major domain concepts and their most common representations, and are not overly detailed. It is not within the scope of this chapter to develop large, complete, and exhaustive ontologies for the specific domains as we merely explore the functionalities of our language by means of concepts within a particular financial or political context. The developed ontologies allow domain experts to annotate texts with common concepts from finance and politics, and to recognize frequently occurring financial and political events.

Our financial ontology contains a small subset of commonly used, well-known, financial entities. Examples of ontology concepts are: companies, products, persons, currencies, CEOs, etc. These concepts have associated lexical representations, e.g., the CEO concept has associated ‘*CEO*’, ‘*Chief Executive Officer*’, ‘*Chief Executive*’, etc. The ontology consists of 65 classes, 18 object properties, 11 data properties, and 1,167 individuals, which can be used for annotation and event detection.

The ontology that is used for event discovery in political news items is also a high-level ontology, yet considerably smaller than the financial ontology. Our political ontology contains 14 classes, 12 object properties, 5 data properties, and 391 individuals. Most individuals are associated with countries. Also, we included many lexical representations of politics-related nouns and verbs, e.g., those linked to elections, provocations, meetings, etc.

For each data set, three domain experts manually annotate the events and relations that we take into account in our evaluation, based on an inter-annotator agreement of at least 66% (i.e., at least two out of three annotators should agree). During the evaluation we focus on the extraction of ten events and relations from the financial domain and ten events and relations from the political domain. Each of these events are described in Tables 4.2 and 4.3, by a name, subject, relation, and an optional object. Based on the events and relations that exist in the news items in the training sets, we let three domain experts construct a set of information extraction rules, where we take the conjunction of

Name	Subject	Relation	Object
CEO	[kb:Company]	kb:hasCEO	[kb:Person]
Product	[kb:Company]	kb:hasProduct	[kb:Product]
Shares	[kb:Company]	kb:hasShareValue	literal
Competitor	[kb:Company]	kb:hasCompetitor	[kb:Company]
Profit	[kb:Company]	kb:hasProfit	literal
Loss	[kb:Company]	kb:hasLoss	literal
Partner	[kb:Company]	kb:hasPartner	[kb:Company]
Subsidiary	[kb:Company]	kb:hasSubsidiary	[kb:Company]
President	[kb:Company]	kb:hasPresident	[kb:Person]
Revenue	[kb:Company]	kb:hasRevenue	literal

Table 4.2: Relations and events for the financial domain, used for evaluation purposes.

Name	Subject	Relation	Object
Election	[kb:Person]	kb:isElectedAs	[kb:Function]
Visit	[kb:Person]	kb:visits	[kb:Country]
Sanction	[kb:Country]	kb:sanctions	[kb:Country]
Join	[kb:Country]	kb:joins	[kb:Union]
Resignation	[kb:Person]	kb:resignsFrom	[kb:Function]
Investment	[kb:Country]	kb:investsIn	[kb:Country]
Riots	[kb:Country]	kb:hasRiots	N/A
Collaboration	[kb:Country]	kb:collaboratesWith	[kb:Country]
Provocation	[kb:Country]	kb:provokes	[kb:Country]
Help	[kb:Country]	kb:helps	[kb:Country]

Table 4.3: Relations and events for the political domain, used for evaluation purposes.

the three constructed rule sets. The constructed rules are subsequently matched to the news items in the test sets, in order to measure the performance.

In our experiments, for each rule group we compare the performance of lexico-syntactic patterns (our baseline) to the performance of lexico-semantic patterns written in HIEL and in JAPE in terms of construction time (i.e., efficiency) and in terms of precision and recall (i.e., expressivity). The latter two measures are often employed in the information extraction field, i.e., *precision P* and *recall R*. These measurements are defined as follows:

$$P = \frac{|\{\text{Relevant}\} \cap \{\text{Found}\}|}{|\text{Found}|}, \quad (4.2)$$

$$R = \frac{|\{\text{Relevant}\} \cap \{\text{Found}\}|}{|\text{Relevant}|}, \quad (4.3)$$

where *Relevant* is the set of relevant annotations (events) and *Found* is the set of found annotations. There is a trade-off between precision and recall, and hence we compute the F_1 measure. The F_1 measure is applied to compute an even combination, i.e., the harmonic mean of precision and recall:

$$F_1 = \frac{2 \times P \times R}{P + R}. \quad (4.4)$$

We measure the rule creation times (excluding reading texts, executing rules, etc.) by averaging the individual rule set creation times of our domain experts. We evaluate the average time it takes for the F_1 measures to become equal to or higher than 0.5. Such a value would be large enough to rule out randomness, as the F_1 measure for a random classifier (based on prior occurrence probability) is a lot less than 0.5 due to the fact that events are seldomly occurring in a news item (when comparing the likelihood of a specific event occurrence with the absence of a specific event with respect to a possible event word sequence in a news item). In theory, creating patterns with an F_1 performance of 0.5 should be manageable within a reasonable amount of time. Additionally, with F_1 scores of 0.5, one avoids the risk of overfitting patterns to a specific data set.

We hypothesize that the creation of well-performing lexico-syntactic rule groups requires more time than the creation of the equivalent lexico-semantic ones. In a second experiment, rule quality, indicated by the precision, recall, and F_1 measures is evaluated for lexico-syntactic, HIEL, and JAPE rule groups given a fixed time in which our domain experts are allowed to create and improve the individual rules. We allow the domain experts to improve their (HIEL and JAPE) lexico-semantic rule groups up until the time it took for creating the equally performing lexico-syntactic rule groups.

4.6.2 Evaluation Results

The construction times presented in Tables 4.4 and 4.5 confirm our hypothesis that the creation of lexico-syntactic rules requires more time than the creation of equally performing lexico-semantic rules, both in HIEL and in JAPE. The tables display rule group creation times in seconds for the lexico-syntactic and lexico-semantic variants, which are obtained on our test sets while aiming for an F_1 score of at least 0.5. For our financial data set, on average, equally well-performing lexico-semantic rule groups are created up to 5 to 70 times faster than their lexico-syntactic counterparts. For JAPE patterns,

Name	HIEL		JAPE
	Lex-Syn	Lex-Sem	Lex-Sem
CEO	8,424	281	738
Product	9,428	132	312
Shares	2,403	648	703
Competitor	9,116	133	850
Profit	1,923	416	1,027
Loss	5,991	313	589
Partner	4,924	185	474
Subsidiary	6,620	776	1,851
President	4,239	179	722
Revenue	5,317	498	798
Overall	5,839	356	806

Table 4.4: Creation times (in seconds) of lexico-syntactic and lexico-semantic rule groups in HIEL, and lexico-semantic rule groups in JAPE, using the financial test set ($F_1 \geq 0.5$).

Name	HIEL		JAPE
	Lex-Syn	Lex-Sem	Lex-Sem
Election	1,517	232	689
Visit	4,238	543	913
Sanction	4,013	419	1,247
Join	3,986	297	405
Resignation	1,259	366	540
Investment	5,162	781	2,304
Riots	1,734	306	451
Collaboration	1,103	137	719
Provocation	1,428	530	828
Help	1,987	211	362
Overall	2,643	382	846

Table 4.5: Creation times (in seconds) of lexico-syntactic and lexico-semantic rule groups in HIEL, and lexico-semantic rule groups in JAPE, using the political test set ($F_1 \geq 0.5$).

creation times are considerably lower than for lexico-syntactic rules, yet they are higher than those for HIEL lexico-semantic patterns. Additionally, for our political data set we observe similar results, although the measured differences are regularly smaller.

The major cause of the construction time reduction that is measured when switching from lexico-syntactic to lexico-semantic patterns lies within the fact that concepts used in HIEL and JAPE lexico-semantic rules, e.g., persons and companies, are conveniently described in an ontology (containing classes, instances, and their associated lexical representations), thus enabling easy reuse. For lexico-syntactic rules however, it is difficult and cumbersome to create rules that distinguish names of persons from companies, products, months, days, etc. Additionally, the verbosity of lexico-syntactic rules and the use of literals to exclude common words (e.g., months) contribute to a considerable amount of extra creation time.

Let us consider a rule that extracts provocation events, where one country provokes another country. When solely utilizing lexico-syntactic elements within the extraction pattern, one would need to intelligently combine lexicographic and orthographic categories. For instance, a country could be defined as a series of nouns and adjectives that contain capitals, i.e.:

```

1   (
2     (JJ | NNS | NNP | NNPS | NN) &
3     (upperInitial | allCaps | mixedCaps)
4   ) +

```

matching phrases like ‘*The Netherlands*’, ‘*Mongolia*’, ‘*United Arab Emirates*’, etc. Additionally, this could be extended so that it would also match strings like ‘*U.S.*’ by adding an extra condition, resulting in:

```

1   (
2     (
3       (JJ | NNS | NNP | NNPS | NN) &
4       (upperInitial | allCaps | mixedCaps)
5     )
6     ('.' NNP '.'?)?
7   ) +

```

However, finding the right combination of nouns and conditions in order to match countries and not other named entities such as persons, companies, products, etc., is a tedious task. An example of a lexico-syntactic rule that can be employed for provocation discovery is:

```

1 PREFIX kb:"http://www.hermes.com/knowledgebase.owl#"
2 ($sub, kb:provokes, $obj) :-
3     $sub := (
4         (
5             (JJ | NNS | NNP | NNPS | NN) &
6             (upperInitial | allCaps | mixedCaps)
7         )
8         ('.' NNP '.')?
9         )+
10        (!'.' & !'(' & !')' & !'-' ){0,3}
11        ('angers' | 'angered' | 'accuses' | 'accused' |
12         'insult' | 'insulted' | 'provokes' | 'provoked' |
13         'threatens' | 'threatened')
14        (!'.' & !'(' & !')' & !'-' ){0,3}
15        $obj := (
16            (
17                (JJ | NNS | NNP | NNPS | NN) &
18                (upperInitial | allCaps | mixedCaps)
19            )
20            ('.' NNP '.')?
21        )+ ;

```

Here, the subject and object are defined as series of capitalized nouns, possibly representing countries. Additionally, verbs related to provocation are required. These are enumerated as literals. Finally, the pattern allows up to three non-punctuation tokens in between the countries and the verb.

When replacing lexical categories and literals with concepts stemming from our political ontology, we obtain the following lexico-semantic rule in HIEL:

```

1 PREFIX kb:"http://www.hermes.com/knowledgebase.owl#"
2 ($sub, kb:provokes, $obj) :-
3     $sub := ([kb:Country] | [kb:Continent] | [kb:Union])
4     (!'.' & !'(' & !')' & !'-' ){0,3}
5     (kb:toAnger | kb:toAccuse | kb:toInsult |
6      kb:toProvoke | kb:toThreaten)
7     (!'.' & !'(' & !')' & !'-' ){0,3}
8     $obj := ([kb:Country] | [kb:Continent] | [kb:Union]) ;

```

The rule is much cleaner and takes considerably less effort to write. As concepts like countries, continents, and unions are conveniently described in the ontology, the user merely needs to refer to them and avoids the hassle of trying to find optimal combinations of lexicographic and orthographic categories, keywords, etc. Moreover, lexico-semantic rules exploit the `typeOf` hierarchy, i.e., because of the inference that can be applied to

ontological concepts, the user can suffice with using concepts like `kb:Country`, instead of their individuals like `kb:US`, `kb:UK`, etc., that have associated lexical representations.

Even though JAPE is more expressive than HIEL as it supports templates (macros) as well as the usage of any Java code – which is useful for removing temporary annotations, percolating and manipulating features from previous annotations, etc. – HIEL rules offer more accessibility to the user. Let us consider the following rule, which is an exact JAPE copy of our previously introduced HIEL rule:

```

1 Rule: Geo_provokes_Geo
2 (
3   (
4     {Lookup.classURI == "Country"} |
5     {Lookup.classURI == "Continent"} |
6     {Lookup.classURI == "Union"}
7   ):sub
8   ({!Token.string ==~ "[.() -]"} [0,3]
9   (
10    {Lookup.URI == "toAnger"} |
11    {Lookup.URI == "toAccuse"} |
12    {Lookup.URI == "toInsult"} |
13    {Lookup.URI == "toProvoke"} |
14    {Lookup.URI == "toThreaten"}
15  )
16  ({!Token.string ==~ "[.() -]"} [0,3]
17  (
18    {Lookup.classURI == "Country"} |
19    {Lookup.classURI == "Continent"} |
20    {Lookup.classURI == "Union"}
21  ):obj
22 )
23 :match --> :match.provokes =
24   {sub = :sub.Lookup.propertyValue,
25    obj = :obj.Lookup.propertyValue}

```

Even without employing the extra features that JAPE rules offer, we already obtain a rule that is more verbose. Therefore, this rule takes considerably longer to write than lexico-semantic HIEL rules. On the other hand, due to the availability of ontology concepts also the creation of lexico-semantic JAPE rules requires less effort than constructing plain lexico-syntactic rules.

In Table 4.6, the experimental results of lexico-semantic rules on the test set are displayed for the financial data set. After allowing the domain experts to improve the lexico-semantic rules written in HIEL up until the time it took for creating the equally performing lexico-syntactic rules (e.g., [2,403–648=] 1,755 extra seconds for shares dis-

Name	Lex-Syn HIEL			Lex-Sem HIEL			Lex-Sem JAPE			
	P	R	F_1	+	-	P	R	F_1	+	-
CEO	0.522	0.600	0.558	69	24	0.897	0.867	0.881	58	8
Product	0.667	0.412	0.509	84	80	0.861	0.772	0.814	122	31
Shares	0.429	0.667	0.522	70	15	0.900	0.800	0.847	40	9
Competitor	0.533	0.480	0.505	45	26	0.760	0.760	0.760	50	12
Profit	0.750	0.455	0.566	20	18	0.880	0.667	0.759	25	11
Loss	0.647	0.407	0.500	17	16	0.813	0.482	0.605	16	14
Partner	0.386	0.739	0.508	44	6	0.800	0.870	0.833	25	3
Subsidiary	0.750	0.391	0.514	24	28	0.906	0.630	0.744	32	17
President	0.433	0.591	0.500	30	9	0.667	0.636	0.651	21	8
Revenue	0.643	0.409	0.500	14	13	0.714	0.682	0.698	21	7
Overall	0.549	0.494	0.520	417	235	0.839	0.741	0.787	410	120
							0.853	0.575	0.687	313
								0.687	197	

Table 4.6: Results of lexico-syntactic and lexico-semantic rule groups on the financial test set (within fixed time), displaying precision (P), recall (R), and F_1 scores, as well as the number of items found (+) and the number of items missed (-).

Name	Lex-Syn HIEL			Lex-Sem HIEL			Lex-Sem JAPE			
	P	R	F_1	+	-	P	R	F_1	+	-
Election	0.462	0.546	0.500	13	5	1.000	0.818	0.900	9	2
Visit	0.677	0.404	0.506	31	31	0.703	0.500	0.584	37	26
Sanction	0.526	0.492	0.509	57	31	0.786	0.721	0.752	56	17
Join	0.722	0.406	0.520	18	19	0.813	0.813	0.813	32	6
Resignation	0.909	0.385	0.541	11	16	0.800	0.923	0.857	30	2
Investment	0.521	0.481	0.500	48	27	0.778	0.673	0.722	45	17
Riots	0.520	0.520	0.520	50	24	0.707	0.820	0.759	58	9
Collaboration	0.425	0.654	0.515	40	9	0.714	0.769	0.741	28	6
Provocation	0.773	0.425	0.548	22	23	0.786	0.571	0.662	28	18
Help	0.551	0.474	0.509	49	30	0.754	0.807	0.780	61	11
Overall	0.566	0.469	0.513	339	215	0.763	0.716	0.739	384	114
							0.701	0.631	0.664	368
								0.687	149	

Table 4.7: Results of lexico-syntactic and lexico-semantic rule groups on the political test set (within fixed time), displaying precision (P), recall (R), and F_1 scores, as well as the number of items found (+) and the number of items missed (-).

covery), the overall precision and recall are 84% and 74%, respectively, resulting in an F_1 score of approximately 79%. With measured precision, recall, and F_1 scores of 85%, 58%, and 69%, respectively, the lexico-semantic rules that are written in JAPE perform notably better than the lexico-syntactic rules, which have a precision, recall, and F_1 measure of 55%, 49%, and 52%, respectively, yet their performance is consistently worse than the performance of lexico-semantic HIEL rules.

For both lexico-semantic pattern languages, the highest recalls are obtained for CEO, Shares, and Partner relations. This is mainly due to the homogeneous sentence structures related to these relations. Judging from the low recalls, the subsidiary and president relations were harder to discover in the text. This could be caused by overfitted rules, which means that it was difficult to create generic rules on the training set that would match many different instances of these relations. The same can be said for the precision and recall (and hence the F_1 value) of the discovery of a company's loss. Another notable observation is the high number of product relations that are discovered in our data set, which can be explained by the fact that many news items discuss companies and their products.

For our political data set, we observe similar overall performances, as depicted in Table 4.7. Generally, lexico-semantic patterns written in HIEL perform better than those written in JAPE. While we observe a precision and recall of 76% and 72%, respectively, for lexico-semantic HIEL rule sets, JAPE rules measure respective scores of 70% and 63%. With F_1 scores of 74% and 66%, this is still considerably better than the 51% accomplished by the lexico-syntactic rules. High precisions and recalls are observed in rules covering elections, resignations, and riots, as these events can usually be found in non-complex sentences where key terms are closely located near one another. Political visits and provocations suffer from low recall values, caused by the wide structural variety and complexity of sentences denoting these events.

On a side note, within our framework it is relatively straightforward to obtain high recall scores. For instance, it would be likely for a rule such as

```

1 PREFIX kb:"http://www.hermes.com/knowledgebase.owl#"
2 ($sub, kb:hasProduct, $obj) :-
3   $sub:=[kb:Company] _*
4   $obj:=[kb:Product] ;

```

to discover each and every existing product relation. However, there is a tradeoff between high recall and high precision. In order to obtain high scores for both measures (expressed in a high F_1 score), rules need to be far more sophisticated. In texts that contain product

relations, often several different companies are mentioned, which makes it difficult to match only the right product with the right company.

Based on the evaluation results, we can conclude that our proposed language, HIEL, is more easy to use for expressing lexico-semantic patterns than the current state-of-the-art JAPE language. Also, we have shown the superiority of lexico-semantic approaches over lexico-syntactic ones with respect to both precision and recall.

4.7 Evaluation of Automatically Learned Patterns

In order to evaluate the effectiveness of our rule learning approach, we have implemented a test method and built a test environment. First we discuss the evaluation setup in Section 4.7.1, followed by the results, in Section 4.7.2.

4.7.1 Evaluation Setup

To evaluate the performance of our information extraction language and the genetic programming approach to automatic rule learning, we make use of the financial data set introduced in Section 4.6, containing Web news articles from the financial and technology domain originating from various sources, including New York Times, Reuters, Washington Post, and Businessweek. Again, each news item is processed using Hermes, and the learned rules are employed for fact extraction (relations between concepts, i.e., triples that denote an event) within the financial domain.

The financial ontology that serves as a basis is slightly different from the one introduced in Section 4.6. Classes and properties have been pruned, and more individuals are included, resulting in more annotations. The ontology consists of 57 classes, 7 object properties, 5 data properties, and 1,287 individuals, which can be used for annotation and event detection.

Again, three domain experts annotate the documents, while distinguishing between ten different financial relations, such as profits, products, CEOs, and competitors of companies. In order to decrease the amount of subjectivity we use a democratic voting principle for the selection of annotations, meaning two out of three annotators should have proposed the annotation to consider it valid. As displayed in Table 4.8, this results in an average Inter-Annotator Agreement (IAA) of 71% for 1,153 unique annotations among all the relations. The table shows that there is a clear difference between the different relations. For instance, the competitor relation is often subjective and therefore hard to determine whether a clear competitor relationship is stated in the text. The same can

Name	Articles	Sentences	IAA
CEO	161	135	0.83
Product	344	300	0.73
Shares	82	77	0.78
Competitor	157	126	0.62
Profit	68	46	0.72
Loss	56	31	0.67
Partner	61	59	0.63
Subsidiary	115	97	0.63
President	64	58	0.68
Revenue	45	20	0.78
Overall	1153	949	0.71

Table 4.8: Inter-Annotator Agreement (IAA) for each of the considered relations.

be argued for the partner relation, which indicates a partnership between two companies. This is in contrast to, for instance, the CEO relation, which is often indicated by words like ‘*CEO*’, ‘*chief*’, or ‘*chief executive*’.

Furthermore the table shows the number of annotations per relation found by the annotators in the set of news items. While the knowledge experts select subjects and objects appearing in separate sentences, which is shown in the second column of Table 4.8, we make a selection of annotations for which the subject and object appear in the same sentence, displayed in the third column. The reason for doing this, is that restricting it to finding relations in a single sentence speeds up the algorithm significantly, while losing only a small portion of the annotations. In future work we intend to experiment with matching a rule onto several sentences, instead of just one. This may also increase the recall, because it often occurs that the subject and the object lie within a certain range from each other, while such an approach still takes less computation time compared to matching the full news item.

Using a hill-climbing procedure, we optimize our algorithm parameters. When learning rules using the genetic programming algorithm with ramped-half-and-half initialization, tournament selection (with a tournament size of 0.25), and a population size of 100, a tree depth of 3 and a maximum amount of children of 7 yields the best results. Here, the mutation rate and elitism rate are 0.3 and 0.05, respectively, whereas the bloat parameter α equals 0.01, making it only effective for situations where F_1 -measures are approximately the same. The group size equals 10, and in our optimal configuration, we only allow for $T = 50$ generations with the same fitness values. Also, during rule learning, we put an emphasis on precision scores with $\beta = 0.3$ for F_β , i.e., an increase in precision is considered to be more important than an increase in recall.

The quality of automatically and manually created rules is evaluated after 5 hours of processing time. Per relation, we observe precision, recall, and F_1 scores. Annotations are used both for rule learning and for manual rule creation in order to verify the quality of intermediate results and to provide a guidance for rule improvements. Therefore, annotation times are excluded from the analysis.

4.7.2 Evaluation Results

The results of the evaluation are presented in Table 4.9, which underlines that, when compared to a full manual approach to rule creation, the use of genetic programming for rule learning can be useful for the considered relations within our evaluated financial domain. The learned rules are used for extracting relations between subjects and objects (facts), i.e., both subject and object have to be correctly identified, as well as the other components used in the rules. Small errors in classification of individual tokens (words) easily disrupt relation detection. Correct classification of relations thus is less trivial than regular named entity recognition, leading to lower results than one would initially expect (Frasincar et al., 2011a).

For automatic rule learning, the CEO relation performs best with a precision, recall, and F_1 -measure of 90%. In a similar manner rules are learned for the president and product relations. For the latter relation we obtain a rule group with a precision and recall of 79%, yielding a 79% F_1 -measure. For the president relation, we measure a precision and recall of 82% and 79%, respectively, resulting in a slightly higher F_1 -measure of 80%. The

Name	Automatic Learning			Manual Creation			$\Delta\%$
	P	R	F_1	P	R	F_1	
CEO	0.904	0.904	0.904	0.824	0.700	0.757	19.5%
Product	0.788	0.793	0.791	0.862	0.596	0.704	12.3%
Shares	0.939	0.805	0.867	0.530	0.778	0.631	37.5%
Competitor	0.667	0.508	0.577	0.875	0.280	0.424	36.0%
Profit	0.960	0.522	0.676	1.000	0.273	0.429	57.7%
Loss	0.905	0.613	0.731	0.818	0.333	0.474	54.3%
Partner	0.808	0.356	0.494	0.450	0.391	0.419	18.0%
Subsidiary	0.698	0.309	0.429	0.611	0.239	0.344	24.8%
President	0.821	0.793	0.807	0.833	0.455	0.588	37.2%
Revenue	0.900	0.450	0.600	0.455	0.455	0.455	32.0%
Overall	0.839	0.605	0.703	0.726	0.450	0.555	26.6%

Table 4.9: Results of HIEL rule groups in terms of precision (P), recall (R), and F_1 scores for all 10 financial relations (rule groups) after 5 hours of automatic rule learning (left) and manual creation (right).

relation hence performs slightly worse than the CEO relation, even though the structure of text is somewhat similar. This may be caused by the lower number of annotations for the president relation. In addition, we have shown in Table 4.8 that the IAA for this relation is slightly lower compared to the CEO relation.

For the competitor, subsidiary, and partner relations, the precision, recall, and F_1 -measure are lower in comparison with the aforementioned relations, approximately ranging between 40% and 60%. This could be caused by the fact that both the subject and the object of these relations are expected to be of type `kb:Company`, while for other types of relation – e.g., product and CEO – the subject and object are of different types, increasing the importance of finding contextual concepts that specifically describe the relation at hand. Additionally, in retrospect, the structure of the sentences in our data describing such relations is more complex than for other relations. In order to find more suitable patterns, the patterns need to be more complex by, for instance, adding more conjunction and negation operators, with the risk of overfitting. Future work should therefore focus on determining how patterns can be learned from more complex sentences, by for instance pre-analyzing the rules for often returning concepts and increasing the probability of appearance for these concepts during initialization and mutation.

The remaining relations, i.e., loss, profit, revenue, and shares are all data properties, meaning they do not require a concept for the object of the relation. Examples of the data property values are ‘*10.5 million euros*’, ‘*\$12*’, or ‘*53 thousand yen*’. In order to match those values one may need a complex pattern, and hence we use the classification component of Hermes to annotate currency values as a single token. For example, the string ‘*10.5 million euros*’ is annotated with a single annotation, e.g., `kb:CurrencyValue`, which can be used in the information extraction rules. This allows us to treat these data properties in a similar manner as the object properties.

Last, the results for automatic rule generation depicted in Table 4.9 show that among the data properties, the shares relation achieved the highest F_1 -value, i.e., 87%, followed by the loss relation, which measured an F_1 -value of 73%. The profit and sales relations performed slightly worse, resulting in F_1 -measures between 60% and 70%.

Our experiments show that the used fitness function – defined in Equation 4.1 – is expensive because the F_1 -measure has to be calculated for each rule in each generation of a population, and is heavily dependent on available computing power. On our machine, an Intel® 2.66 GHz Core™ i7 920 processor with 6 GB of RAM, jobs finished within 5 hours each. On average, the generation of a rule group representing a relation takes approximately 4 and a half hours. The largest amount of time needed for one rule group was 5 hours, whereas the smallest amount of time required was 3 and a half hours.

We also let a domain expert create rules manually for 5 hours per rule group on the same machine to ensure a fair comparison of our automatic system with the manual creation of rules. Again, most time is consumed by evaluating rules, yet a manual approach is less efficient. Where the genetic programming approach generates precision, recall, and F_1 -values of 84%, 61%, and 70%, respectively, on average, the manually created rule groups show lower performances. For manual rule creation, the resulting F_1 -values are on average about 27% lower (displayed under $\Delta\%$ in the rightmost column of Table 4.9). Hence, within the same amount of time (i.e., 5 hours per rule group), a domain expert manually writing rules would end up with worse performing rules than an automated genetic programming-based approach. We do not question the potential quality of the rules manually created by the experts when allowing for more time, yet within the limited amount of time advantages of automatic generation are clearly shown. We do, however, observe similar performance patterns as have been described above.

The largest improvements (up to 58%) we observe for relations that involve data properties that deal with more complex constructions (e.g., using datatype variants), which are cumbersome for human experts to include in their rules, hence leading to lower recall. For example, loss and profit relations involve complex sentences with currencies, which have many different variants in our data set. On the other hand, rule groups that cover many structurally homogeneous examples for which the subject and object are concepts having different types, e.g., the groups associated with product and CEO relations, show improvements as low as 12%, as these are straightforward to implement for domain experts, thus diminishing the need for automation.

For the domain expert, the actual writing takes up a few percent of the total time (5 to 10 minutes). A considerable amount of time is used for reading news messages, analyzing matched patterns, verifying results, etc., which explains the differences with the results in Section 4.6.2, where only the writing times are considered. Perfecting rules takes up increasingly more time, as one needs to abstract away from examples in the training set. When increasing the training set size, it becomes nearly impossible for domain experts to keep up with a genetic programming-based approach, underlining the added value for automatic rule generation for detecting complex semantic relations in large data sets.

4.8 Conclusions

As structuring data on the Web is a tedious and time-consuming process, in this chapter, we proposed a rule-based method to extract relations and events in news articles. The contribution to the existing body of knowledge is threefold.

First, our proposed method relies on the Hermes Information Extraction Language (HIEL), i.e., a lexico-semantic pattern language that not only makes use of lexical and syntactical elements, but also employs ontology concepts and relations. These patterns are based on regular expressions, which enhance the expressivity of the rules. In this chapter, we have provided a formal syntax for the lexico-semantic rules. Second, in order to show how the proposed rule-based extraction method can be applied in practice, we have implemented the approach in the Hermes News Portal (HNP) as the Hermes Information Extraction Engine (HIEE) plug-in. Combined with standard text preprocessing tasks performed by the GATE framework, as well as a central knowledge base expressed in an OWL ontology, events and relations that occur in news items are extracted. Last, as manual construction of rules often proves to be time-consuming, we have additionally investigated a genetic programming-based approach for rule learning (based on financial news). Genetic programming approaches provide the user with insight into how rules are learned and usually find adequate solutions within a reasonable amount of time.

In order to assess the performance of our proposed method, we have evaluated the implementation by building rules and measuring the performance of the extraction of events and relations by using these rules. On two separate data sets and corresponding ontologies from the financial and political domains, this resulted in a precision of approximately 80% and a recall of 70%, as the lexico-semantic patterns are superior to lexico-syntactic patterns with respect to expressivity. Additional experiments show that, when compared to lexico-semantic rules in JAPE, lexico-semantic HIEL rules obtain higher precision and recall scores than their JAPE equivalents.

Furthermore, our experiments showed that creating lexico-semantic rules requires significantly less time than creating equally performing lexico-syntactic rules, as lexico-semantic rule group creation times were in general one degree of magnitude smaller than lexico-syntactic rule group creation times. We argue that lexico-syntactic rules require more development time because of the larger amount of effort needed for entering the individual literals, resulting in low precision. Also, lexico-semantic rules exploit the inference capabilities of ontologies. This underlines the advantage of using lexico-semantic rules. Moreover, we have demonstrated that lexico-semantic HIEL rules are less verbose than their JAPE equivalents, resulting in less construction time and contributing to higher precision and recall values.

Moreover, our rule learning system performs good in terms of recall and precision, and hence also yields good F_1 -values of 70% across all considered financial relations. Our experiments show that compared to information extraction rules constructed by expert users, we are able to find rules that yield a higher F_1 -value (i.e., 27% higher on average)

after the same amount of time (i.e., 5 hours). A frequently encountered problem for the genetic programming approach is that the quality of the initial population is too low, because the probability that the right concepts are initially chosen becomes smaller as the total number of concepts in the knowledge base increases.

While we have focused on finding new information and identifying events and relations in news articles, as future research we suggest to focus on (semi-)automatically processing the information that was found and updating the ontology (Sangers et al., 2012b). Additionally, in our approach, we can only extract one triple per rule (the left-hand side of the rule), while events often consist of more than a subject, predicate, and an object. For instance, time can play a role in the event. Also we want to increase the expressivity of our lexico-semantic patterns by making use of the relationships stored in the ontology, or going one step further by employing the expressivity of one-dimensional SPARQL queries. Moreover, we aim to investigate solutions to the aforementioned rule learning problem regarding the low quality of initial rule populations, e.g., by implementing heuristics and bootstrapping our algorithms. We hypothesize that frequently appearing concepts in a certain domain can be given a higher probability during initialization, in order to increase the quality of the initial population. Moreover, manually derived rules can be useful as well when deployed in the initial population. Also, we plan to extend our rule learning evaluation to also include single rule matching on multiple sentences. Last, an additional direction for future work with respect to our rule learning method is the extraction of other types of information (from different domains than the financial domain, such as the political, medical, and weather domains).

4.A Appendix: Hermes Information Extraction Language Grammar

The Hermes Information Extraction Language (HIEL) that is presented in this chapter can be formally described in the Backus Naur Form (BNF). The non-terminals used in our language, which make use of groups of terminal symbols discussed next, can be described as follows:

```

1 <Start>      ::= <SPACE>* (<Prefix> <SPACE>*)*
2                      <Lhs> <SPACE>* <COL_MIN> <SPACE>*
3                      <Rhs> <SPACE>* <SEMI_COL> <SPACE>*
4 <Prefix>       ::= <PREFIX> <SPACE>+ <Ns> <Url>
5 <Ns>          ::= (<CHAR> | <UNI> | <NUMBER>)+ <COL>
6 <Url>          ::= <DQ> <CHAR>+ <COL> <FSLASH> <FSLASH>

```

```

7          (<CHAR> | <NUMBER> | <DOT> | <COMMA> | <QSTN> |
8          <EXCL> | <SEMI_COL> | <SQ> | <ULINE> | <MIN> |
9          <PLUS> | <EQ> | <PL> | <PR> | <BL> | <BR> |
10         <AND> | <TILDE> | <STAR> | <AT> | <PERC> |
11         <DOLLAR> | <FSLASH>)*
12         [<HASH>] <DQ>
13 <Lhs>      ::= <PL> <SPACE>*
14           <LhsElement> <SPACE>* <COMMA>
15           <SPACE>* <LhsElement> <SPACE>*
16           [<COMMA> <SPACE>* <LhsElement>]
17           <SPACE>* <PR>
18 <LhsElement> ::= (<LhsVariable> | <LhsProp>)
19 <LhsVariable> ::= <DOLLAR> <Name>
20 <LhsProp>    ::= <Ns> <Name>
21 <Class>     ::= <BL> <Ns> <Name> <BR>
22 <Indv>      ::= <Ns> <Name>
23 <Name>       ::= (<CHAR> | <UNI> | <NUMBER> | <MIN> | <ULINE>)+
24 <Element>   ::= (<String_Lit> | <Syn> | <Orth> | <Class> |
25           <Indv>)
26 <String_Lit> ::= (<String_Lsq> | <String_Ldq>)
27 <String_Lsq> ::= <SQ> <Seq> <SQ>
28 <String_Ldq> ::= <DQ> <Seq> <DQ>
29 <Seq>        ::= (<NUMBER> | <CHAR> | <UNI> | <ESC> | <SPACE>)+
30 <Syn>        ::= <SYN>
31 <Orth>       ::= <ORTH>
32 <Rhs>        ::= [<Label>] (<RhsP> | <RhsCP>)
33           (<SPACE>+ [<Label>] (<RhsP> | <RhsCP>))*+
34 <Label>      ::= <DOLLAR> <Name> <COL_EQ>
35 <RhsP>       ::= (([<EXCL>] <Element> [<RepOp>]) |
36           (<ULINE> [<RepOp>]))
37 <RhsCP>      ::= [<EXCL>] <PL> <SPACE>* (<RhsP> | <RhsCP>)
38           <SPACE>* ((<AND> | <OR>) <SPACE>*
39           (<RhsP> | <RhsCP>) <SPACE>*)*
40           <PR> [<RepOp>]
41 <RepOp>      ::= ((<QSTN> | <STAR> | <PLUS>) |
42           (<AL> <NUMBER> [<COMMA> [<NUMBER>]] <AR>))

```

Next, all terminals – i.e., all literal, elementary symbols that cannot be changed using the grammar rules – used in HIEL are summarized as follows:

```

1 <NUMBER>    ::= ([0-9])+*
2 <CHAR>      ::= ([A-Z] | [a-z])
3 <UNI>        ::= '\u' ([0-9] | [A-F] | [a-f])
4           ([0-9] | [A-F] | [a-f])
5           ([0-9] | [A-F] | [a-f])
6           ([0-9] | [A-F] | [a-f])*
7 <ESC>        ::= ('\' | '\"' | '\\')

```

```

8  <SPACE>      ::= (' ' | '\n' | '\r' | '\n\t')
9  <DOT>         ::= '.'
10 <COMMA>       ::= ','
11 <QSTN>        ::= '?'
12 <EXCL>        ::= '!'
13 <COL>          ::= ':'
14 <SEMI_COL>    ::= ';'
15 <SQ>           ::= "'"
16 <DQ>           ::= "\""
17 <ULINE>        ::= '_'
18 <MIN>          ::= '-'
19 <PLUS>         ::= '+'
20 <EQ>           ::= '='
21 <COL_EQ>       ::= ':='
22 <COL_MIN>     ::= ':-'
23 <PL>           ::= '('
24 <PR>           ::= ')'
25 <AL>           ::= '{'
26 <AR>           ::= '}'
27 <BL>           ::= '['
28 <BR>           ::= ']'
29 <AND>          ::= '&'
30 <OR>           ::= '|'
31 <TILDE>        ::= '~'
32 <STAR>         ::= '*'
33 <AT>           ::= '@'
34 <PERC>         ::= '%'
35 <DOLLAR>       ::= '$'
36 <HASH>         ::= '#'
37 <FSLASH>       ::= '/'
38 <PREFIX>       ::= 'PREFIX'
39 <SYN>          ::= ('CC' | 'CD' | 'DT' | 'EX' | 'FW' | 'IN' | 'JJ' |
40                                | 'JJR' | 'JJS' | 'JJSS' | 'LS' | 'MD' | 'NN' | 
41                                | 'NNP' | 'NNPS' | 'NNS' | 'NP' | 'NPS' | 'PDT' | 
42                                | 'POS' | 'PP' | 'PRPR$' | 'PRP' | 'PRP$' | 'RB' | 
43                                | 'RBR' | 'RBS' | 'RP' | 'SYM' | 'TO' | 'UH' | 
44                                | 'VBD' | 'VBG' | 'VBN' | 'VBP' | 'VB' | 'VBZ' | 
45                                | 'WDT' | 'WP$' | 'WP' | 'WRB')
46 <ORTH>         ::= ('upperInitial' | 'allCaps' | 'lowerCase' | 
47                                | 'mixedCaps')

```


Chapter 5

Event-Driven Ontology Updating[§]

ONTOLOGIES, as reliable resources in decision making processes, need to be accurate and up-to-date. For this purpose, ontologies have to be maintained regularly. Manual updating is tedious and time-consuming, therefore we propose an event-driven automated ontology updating approach. The Ontology Update Language (*OUL*) and our proposed extensions are inspired by the existing SQL-triggers mechanism and make use of SPARQL and SPARQL/Update statements. We propose different execution models, providing flexibility with respect to the update process. As a proof-of-concept, we implement the language and its execution models in the *Hermes News Portal (HNP)*, an ontology-based news personalization service.

[§]This chapter is based on the conference publication “J. Sangers, F. Hogenboom, and F. Frasincar. Event-Driven Ontology Updating. In X. S. Wang, I. F. Cruz, A. Delis, and G. Huang, editors, *13th International Conference on Web Information System Engineering (WISE 2012)*, volume 7651 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2012.”

5.1 Introduction

One of the most important driving factors for information in today's society is news. Every day, millions of people try to keep up-to-date with the latest developments by reading news items. Next to television and newspapers, the World Wide Web has become a good alternative for people to keep track of the state of the world. Developments in the real world – described in news items – influence a variety of activities, ranging from individual daily activities such as buying products to companies' long-term business strategies. Lately, there has been an increasing amount of effort put into automatically processing news data by extracting important information. Applications that make use of this information are plentiful, e.g., automated stock agents that keep track of financial news to exploit extracted knowledge on the stock market, news personalization services that provide users with information that matches user interests, etc.

Traditionally, news is presented as plain text and can be characterized as unstructured data, making it hard for computer systems to interpret it. With the Semantic Web, the World Wide Web Consortium (W3C) provides a framework to add structure to data through the usage of the Web Ontology Language (OWL) (Bechhofer et al., 2004). By means of ontologies, domain specific knowledge can be represented by creating concepts and relations between these concepts. The relations are established by defining triples that consist of a subject, a predicate, and an object.

With structured data, information can be easily extracted, and interoperability between computer systems is stimulated. This information, often described using ontologies, is used as an information source that influences the systems' actions. Due to the non-static nature of our society, the information that reflects the real world at any given time has to be updated regularly. Traditional data sources like relational databases have mechanisms for automatic updates. However, a principled way of automatic ontology updating does not yet exist. This forces domain experts to manually update ontologies, which is a tedious, repetitive, error-prone, and time-consuming job.

Numerous applications, such as the Hermes News Portal (HNP) (Frasincar et al., 2009) – an ontology-based news personalization service – take advantage of Web news items by exploiting their information through ontology matching. As a classification and querying tool, it is important that the ontology contains up-to-date information. However, tools like the HNP often lack an update language for maintaining underlying ontologies and would therefore benefit from an ontology update language. The Ontology Update Language (OUL) (Lösch et al., 2009) is such an update language, and alleviates the process of manual updating ontologies by defining sets of SPARQL/Update (Seaborne et al., 2008) rules. It is

based on an automatic update mechanism and operates using the Event-Condition-Action model, where event occurrences trigger actions through handlers and preconditions are assessed. However, OUL has limited flexibility in execution models, and hence there is no support for a fully automated update mechanism. Therefore, in this chapter we propose OULx, an extension to OUL language supporting various additional execution mechanisms inspired from active databases. As a proof-of-concept, we implement the language and its execution models in the HNP.

The remainder of this chapter is structured as follows. First, Section 5.2 discusses related work. Next, Sections 5.3 and 5.4 elaborate on the proposed language and execution model extensions. An implementation is discussed in Section 5.5 and OULx is evaluated in Section 5.6. Last, Section 5.7 concludes the chapter and provides directions for future research.

5.2 Related Work

Due to the recent explosion in (meta-)data representation technologies, information can be described in many ways. One way to do this is by making use of relational databases, which store information in tables related to each other. Additionally, the eXtensible Markup Language (XML) (Bray et al., 2008) can describe the information in a tree-structure, a common way for transportation of information between systems. Last, semantic languages for storing information exist. The Resource Description Framework (RDF) (Brickley and Guha, 2004) adds meaning to data by using triples and can be serialized in XML. OWL extends RDF with the possibility to express additional constraints and is often used as an ontology representation language.

Commonly used languages for retrieving information from sources are the Structured Query Language (SQL) (Chamberlin and Boyce, 1974) for relational databases, XPath (Clark and DeRose, 1999) and XQuery (Boag et al., 2010) for XML documents, and SPARQL (Prud'hommeaux and Seaborne, 2008) for RDF and OWL. Although SQL is mainly used for querying information from tables, extra functionalities have been added to it, such as the creation, alteration, and removal of tables. These statements can be executed individually, but can also be used in combination with SQL triggers. These triggers react on predefined events based on an Event-Condition-Action model and execute SQL statements either immediately or deferred if a condition is met, hereby creating an automated way of updating relational databases.

Updating XML documents can be realized with XUpdate (Laux and Martin, 2000), and updating ontologies is usually done with SPARQL/Update statements (Seaborne

et al., 2008). These statements are similar to SPARQL queries, though specifically designed for updating ontologies. Due to the complexity of ontology updating caused by dependencies and physical distributions, we need a principled approach for automatic ontology updating. The Ontology Update Language (OUL) (Lösch et al., 2009) is a blend of active (database) triggers and SPARQL/Update statements, which updates ontologies in an event-driven manner. By defining so-called *changehandlers*, specific ontology change events can be caught and handled individually.

Despite the convenient representation aspects of OUL inspired from active database triggers, the usage of SPARQL and SPARQL/Update, and the implementation of preconditions, the language lacks several key features. First, OUL does not support negation and namespaces. Second, chaining of triggers (changehandlers) is not possible. The changehandlers do not react on actions of other changehandlers. In order to trigger a changehandler, the user has to manually execute an update. Third, there is no differentiation between the order of execution of changehandlers' actions, i.e., there is no distinction between immediate (i.e., once a changehandler is matched, it is executed) and deferred (i.e., the actions are executed all at once after matching the changehandlers and collecting the actions) executions. Fourth, only the first matching changehandler is executed. Hence, when an event occurs, each changehandler is matched against the event; the first changehandler that matches, is handled. Additionally, when updates are triggered and executed, new updates could be triggered, requiring another update cycle. This kind of execution looping is currently not supported.

5.3 OUL Syntax

Atomic ontology update actions can be executed using SPARQL/Update statements. However, multiple ontology change actions are often required. These actions are hard to express in one single SPARQL/Update statement and can not be edited easily. Therefore, complex ontology updates should be performed as a sequence of atomic SPARQL/Update statements executed in a specific order. The Ontology Update Language (OUL) (Lösch et al., 2009) is based on the automatic update mechanism in active databases: SQL-triggers. Using an Event-Condition-Action model, a list of ontology update actions are performed on event occurrence. This method, using triggers (called changehandlers here), however, does not support a fully automated ontology update process. OUL does feature a dynamic update process using an existing RDF update language, and hence we extend this language in such a way that no human intervention is needed for multiple updates.

OWL makes use of changehandlers that perform SPARQL/Update actions whenever a certain change event (represented as an RDF-graph that is either added to or deleted from the ontology) occurs. If we want to perform particular actions, whenever such triple is added to or deleted from the ontology, we can specify them in a changehandler. Each changehandler has a general form as:

```

1 CREATE CHANGEHANDLER <name>
2 FOR <changerequest>
3 AS
4   [ IF <precondition>
5     THEN ] <actions>
```

which is analogous to active database triggers. When the *changerequest* matches the change event, a *precondition* on the ontologies is checked. If this precondition is met or if no precondition has been defined, a list of *actions* will be executed. In contrast to active databases, ontology updates do not require SQL statements, but events, conditions, and actions have to be defined using SPARQL and SPARQL/Update statements.

5.3.1 Requesting Changes

OWL defines two different types of changerequests, i.e., insertion and deletion of information. The `add` and `delete` keywords distinguish between the two different types and every changerequest is further defined by a WHERE-clause of a SPARQL SELECT query. The syntax is defined as:

```

1 <changerequest> ::= add [unique] (<SPARQL>
2                               | delete [unique] (<SPARQL>)
3 <SPARQL>      ::= WHERE clause of a SPARQL SELECT query
```

When all the triples in the query can be deduced from a change event and the event-type matches the changerequests' type, the changerequest is matched. The set of bindings that are returned from the query can be reused later in the `AS`-clause of the changehandler definition. A `unique` property can be used to state whether only one single binding is required. Whenever this property is set, changerequests will not match when their query returns multiple bindings.

5.3.2 Preconditions

Whenever a changerequest matches, also optional preconditions defined in the changehandler have to be met so that the actions are executed. In contrast to the changerequest,

which is used to match the occurring event, the precondition is used to check the current state of the ontology. Three different types of preconditions can be used. First, `contains` checks whether the ontology contains a set of triples. Second, `entails` checks whether the ontology entails a set of triples, i.e., using inferencing it can be concluded that the statement is logically entailed by the ontology. Third, `entailsChanged` checks whether the direct application of the requested change leads to an ontology which entails a set of triples.

Conditions can be combined by `and`- or `or`-operators and can be nested as well. Each precondition results in a set of bindings and the `and`- and `or`-operators perform join and union operations on the resulting bindings. The syntax for the precondition is defined as follows:

```

1 <precondition> ::= contains(<SPARQL>
2                                | entails(<SPARQL>)
3                                | entailsChanged(<SPARQL>)
4                                | (<precondition>
5                                | <precondition> and <precondition>
6                                | <precondition> or <precondition>
7 <SPARQL>      ::= WHERE clause of a SPARQL SELECT query

```

5.3.3 Actions

When the changerequest is matched and the precondition is met, a list of actions is executed. Actions make use of the binding information that resulted from matching the changerequest and the precondition. There are four types of actions, i.e., SPARQL/Update queries, feedback actions that give feedback to the user using text containing bounded variables, `applyRequest` actions that execute the events caught by the changehandler, and last, the `for` actions that iteratively execute a set of actions with binding information from a for-condition:

```

1 <actions>       ::= [<action>] | <action> <actions>
2 <action>        ::= <SPARQL update>
3                                | for( <precondition> ) <actions> end;
4                                | feedback(<text>)
5                                | applyRequest
6 <SPARQL update> ::= MODIFY action (in SPARQL/Update)
7 <text>          ::= string (may contain SPARQL variables)

```

5.3.4 Extensions

In SPARQL it is possible to define prefixes, i.e., labels referring to a namespace. Since in OUL multiple SPARQL WHERE clauses and SPARQL/Update MODIFY clauses may be used, it is necessary to define in each query the used prefixes or to use the full namespaces. The latter provides too much overhead and hence, we propose to define the prefixes for the entire changehandler instead of for every separate SPARQL query.

In OUL, preconditions can be combined by using `or`- or `and`-operators. It is, however, not possible to use negation, something that could be desirable whenever ontologies should not contain certain information. Therefore, we implement negation by allowing the usage of an exclamation mark ('!') to denote negation in OUL. The syntax is altered as follows:

```

1 CREATE CHANGEHANDLER <name>
2 [<prefixes>]
3 FOR <changerequest>
4 AS
5   [ IF <precondition>
6     THEN ] <actions>
7
8 <prefixes>      ::= <prefix> [<prefixes>]
9 <prefix>         ::= <SPARQL prefix>
10 <changerequest> ::= add [unique] (<SPARQL>)
11                   | delete [unique] (<SPARQL>)
12 <precondition>  ::= contains(<SPARQL>)
13                   | entails(<SPARQL>)
14                   | entailsChanged(<SPARQL>)
15                   | (<precondition>)
16                   | <precondition> and <precondition>
17                   | <precondition> or <precondition>
18                   | !<precondition>
19 <actions>        ::= [<action>] | <action> <actions>
20 <action>         ::= <SPARQL update>
21                   | for( <precondition> ) <actions> end;
22                   | feedback(<text>)
23                   | applyRequest
24 <SPARQL prefix> ::= PREFIX statement of a SPARQL query
25 <SPARQL>         ::= WHERE clause of a SPARQL SELECT query
26 <SPARQL update> ::= MODIFY action (in SPARQL/Update)
27 <text>           ::= string (may contain SPARQL variables)
```

A typical example of an OULx changehandler, of which a graphical representation is depicted in Figure 5.1, is displayed on the next page. The handler is specified using the default OUL syntax, but additionally uses the OULx enhancements, i.e., prefixes and negation.

```

1 CREATE CHANGEHANDLER addProductHandler
2 PREFIX kb: <http://www.hermes.com/knowledgebase.owl#>
3 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4 FOR add(?company kb:hasProduct ?product)
5 AS IF (contains(?company rdf:type kb:Company)
6 and !(contains(?product rdf:type kb:Product)))
7 THEN insert data{?product rdf:type kb:Product};
8 applyRequest;

```

The changehandler, which is triggered when adding an item to an ontology of companies and their products, adds missing product data to the ontology in case the company referred to in a request does exist, but the product of interest does not, so that the original request for linking a product to a company can be executed. In the graphical representation, the main elements, i.e., name, prefixes, changerequest, preconditions, and actions, have been highlighted.

5.4 OUL Execution Models

Updating ontologies in an event-driven manner requires an execution model that controls aspects like selecting the proper changehandlers, executing SPARQL queries, and performing changehandler actions. Lösch et al. (2009) provide an execution environment for OUL that allows for ontology updating upon detection of change events in texts. The Ontology Update Manager plays a central role here, as it matches changehandlers based on a changerequest and executes the actions defined in the respective changehandlers. The ontology update specification describes how the ontology can be updated by providing a set of changehandlers.

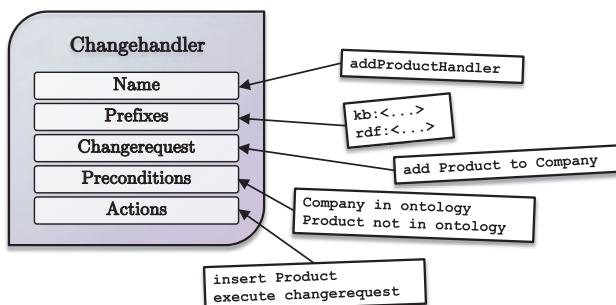


Figure 5.1: A typical OULx changehandler for adding an item to an ontology.

By default, whenever a change event occurs, all changehandlers defined in the ontology update specification are checked upon their changerequest and precondition to determine if the change event can be handled by a specific changehandler. When a changehandler matches a changerequest with a change event and the precondition is met, the original change event is replaced by the actions defined in the matching changehandler. These actions are then stored and executed all at once later on, i.e., in a deferred manner. In situations where multiple changehandlers match a change event and meet their precondition, only the actions of the first matching changehandler are executed. As OUL does not feature chaining of changehandlers, the execution of the actions cannot trigger other changehandlers, implying that immediate execution would have the same results as deferred execution, when executing only the first matched changehandler.

With respect to the original OUL execution model, we propose several extensions. First, inspired by applications in active databases, we extend OUL by adding support for immediate updating, as opposed to deferred updating. Next, in analogy with active databases where triggers can activate other triggers, we add changehandler chaining. Although this does not ensure termination, it enhances the expressivity of the update language and it enables separation of atomic update operations, thereby enabling modularity. Similarly to active databases triggers, methods for automatic termination evaluation can be developed (Ray and Ray, 2001). Additionally, execution looping is added, which is needed in situations where new updates are required after triggering and executing other updates. Last, we update the OUL execution model in a way that it does not only execute the first matching changehandler, but optionally each matched changehandler.

5.4.1 Deferred and Immediate Updating

The original (deferred) execution model of OUL comprises three main steps, which are illustrated in Algorithm 5.1. First, changerequests of all defined changehandlers with respect to the change event are matched and preconditions are verified. Second, actions are collected from the matched changehandlers and SPARQL/Update statements are created. Third and last, the latter statements are applied to the ontology. Note that the method *matchHandlers(...)* is further specified in Algorithms 5.3 (first match) and 5.4 (all matches), and *collectUpdates(...)* is described in Algorithm 5.5.

This execution model can be altered in such a way that immediate updating is performed. This implies that during the collection process, update statements are applied immediately to the ontology. Hence, in contrast to deferred updating, we distinguish between two steps, i.e., changehandler matching and update application. Algorithm 5.2 provides

Algorithm 5.1: Deferred ontology updating (updateOntology).

Task	: update ontology with deferred execution of updates
Input	: ontology O consisting of axioms, change event $op(Ax)$ where $op \in \{add, del\}$ and Ax is a set of axioms
Data	: changehandlers $matchedHandlers$ that match their changerequest and meet their precondition according to the provided change event, list of update actions $updateList$ to be applied to the ontology
Output	: updated ontology O

```

1 // Find matched changehandlers
2 matchedHandlers  $\leftarrow$  matchHandlers( $O$ ,  $op(Ax)$ );
3 // Collect updates from changehandlers
4 updateList  $\leftarrow$  collectUpdates( $O$ ,  $op(Ax)$ , matchedHandlers);
5 // Apply updates to ontology in deferred way
6 foreach update in updateList do
7   | O.apply(update);
8 end
9 return O
```

Algorithm 5.2: Immediate ontology updating (updateOntology).

Task	: update ontology with immediate execution of updates
Input	: ontology O consisting of axioms, change event $op(Ax)$ where $op \in \{add, del\}$ and Ax is a set of axioms
Data	: changehandlers $matchedHandlers$ that match their changerequest and meet their precondition according to the provided change event
Output	: updated ontology O

```

1 // Find matched changehandlers
2 matchedHandlers  $\leftarrow$  matchHandlers( $O$ ,  $op(Ax)$ );
3 // Apply updates to ontology in an immediate way
4 O  $\leftarrow$  applyUpdates( $O$ ,  $op(Ax)$ , matchedHandlers);
5 return O
```

the immediate updating model. Note that the method $matchHandlers(\dots)$ is further specified in Algorithms 5.3 (first match) and 5.4 (all matches), and $applyUpdates(\dots)$, which applies updates, is described in Algorithm 5.6.

The concepts of deferred and immediate updating are also depicted in Figures 5.2(a) and 5.2(b), respectively. For simplicity, we assume the occurrence of a single event, which is matched against a single changehandler. Actions are performed in case preconditions are met (step 1). In case of deferred execution, all actions from the latter changehandler are collected – and optimized – before executing them (steps 2 and 3, respectively), while for immediate execution, the specified actions are performed immediately (step 2). In practice, a set of appropriate changehandlers is often specified, of which multiple can match after assessing their preconditions. Differences between immediate and deferred execution are in general quite small when considering only one changehandler or when executing merely the first matching changehandler of a collection of changehandlers (i.e., the default execution model for OUL), but become more abundant when considering all matching changehandlers.

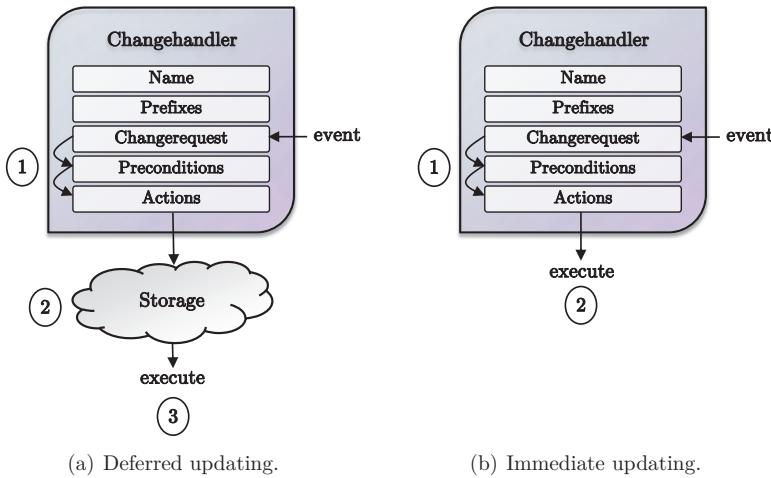


Figure 5.2: Deferred and immediate execution of actions specified in a matched changehandler that has been constructed for a specific event, after successfully verifying the handler’s predefined preconditions.

5.4.2 Matching First and All Changehandlers

There are two distinct ways of matching changehandlers. The OUL execution model proposed by Lösch et al. (2009) returns the first changehandler that matches a changerequest and meets its precondition (Algorithm 5.3). An iterator moves forward through the ontology update specification document until either the end of the document has been reached or a changehandler has been matched to the change event. The matching process returns non-empty binding information which should contain a single binding when a unique keyword is used in the changerequest. For matching preconditions, in case a valid binding is returned, the changehandler is added to the list of matched changehandlers.

However, one could also require multiple changehandlers to be matched. When altering Algorithm 5.3 by changing the loop conditions, we obtain an execution model that returns all matching changehandlers associated with a change event as given in Algorithm 5.4. While in Algorithm 5.3 in line 2 a condition for limiting the list of matched changehandlers is defined, in Algorithm 5.4, this is removed, making it possible to check all changehandlers defined in the ontology update specification and to add every matching changehandler to the resulting list.

Figures 5.3(a) and 5.3(b) depict the matching process of the first and all matching changehandlers, respectively. Here, we consider a set of changehandlers developed specifically for an event. Each of these handlers is evaluated based on its specified preconditions.

Algorithm 5.3: Returning the first matching changehandler (matchHandlers).

```

Task      : collect matching changehandlers
Input     : ontology  $O$  consisting of axioms,
             ontology update specification  $US$  treated as a list of changehandlers,
             change event  $op(Ax)$  where  $op \in \{add, del\}$  and  $Ax$  is a set of axioms.
Data      : changehandler handler that is checked for applicability
Output    : list of matched changehandlers matchingHandlers
1 // While not at document's end and no changehandler has been matched
2 while not  $US.endOfDocument$  and matchingHandlers.count < 1 do
3   // Take the next changehandler
4   handler  $\leftarrow US.nextChangeHandler;$ 
5   // Match the changerequest with the change event
6   matches  $\leftarrow SPARQLmatch(handler.changerequest, op(Ax));$ 
7   // The bindings form the changerequest should not be empty
8   if not matches.isEmpty then
9     // The number of bindings should be 1 when the unique keyword is used
10    if (handler.changerequest.unique and matches.count == 1) or not
11       handler.changerequest.unique then
12         // Substitute variables in the precondition with changerequest bindings
13         instPrecondition  $\leftarrow substitute(handler.precondition, matches.first);$ 
14         // Evaluate the precondition; when this returns any binding, it is met
15         if not evaluate(instPrecondition,  $O$ ).isEmpty then
16           // Add the changehandler to the list
17           matchingHandlers.add(handler);
18         end
19       end
20     end
21   return matchingHandlers

```

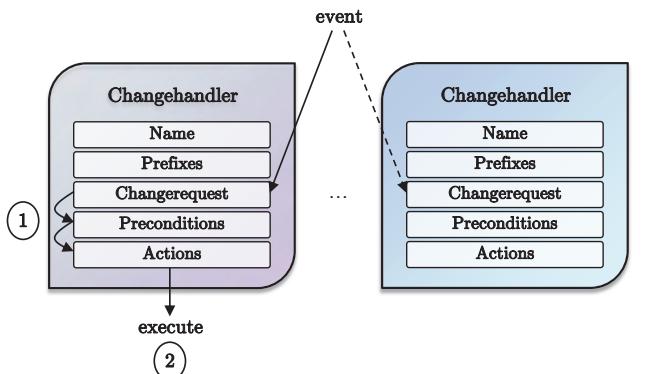
Algorithm 5.4: Returning all matching changehandlers (matchHandlers).

```

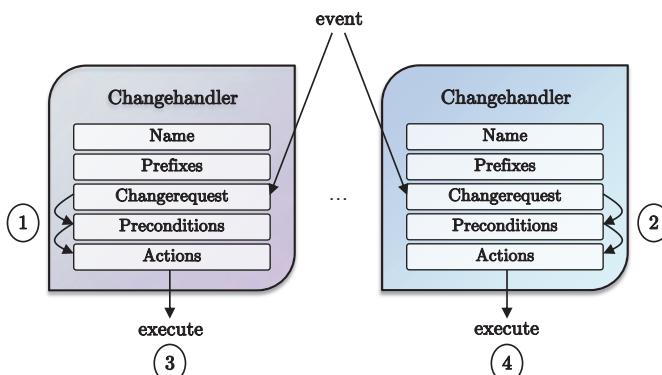
Task      : collect matching changehandlers
Input     : ontology  $O$  consisting of axioms,
             ontology update specification  $US$  treated as a list of changehandlers,
             change event  $op(Ax)$  where  $op \in \{add, del\}$  and  $Ax$  is a set of axioms.
Data      : changehandler handler that is checked for applicability
Output    : list of matched changehandlers matchingHandlers
1 // While not at document's end
2 while not  $US.endOfDocument$  do
3   // Take the next changehandler
4   handler  $\leftarrow US.nextChangeHandler;$ 
5   // Match the changerequest with the change event
6   matches  $\leftarrow SPARQLmatch(handler.changerequest, op(Ax));$ 
7   // The bindings form the changerequest should not be empty
8   if not matches.isEmpty then
9     // The number of bindings should be 1 when the unique keyword is used
10    if (handler.changerequest.unique and matches.count == 1) or not
11       handler.changerequest.unique then
12         // Substitute variables in the precondition with changerequest bindings
13         instPrecondition  $\leftarrow substitute(handler.precondition, matches.first);$ 
14         // Evaluate the precondition; when this returns any binding, it is met
15         if not evaluate(instPrecondition,  $O$ ).isEmpty then
16           // Add the changehandler to the list
17           matchingHandlers.add(handler);
18         end
19       end
20     end
21   return matchingHandlers

```

In case only the first matching changehandler is dealt with (as shown in Figure 5.3(a)), preconditions are evaluated for the first couple of changehandlers until the preconditions are met for the first handler (step 1), of which the associated actions are subsequently executed (step 2). Even though there are more handlers to be evaluated and there is a second handler of which the preconditions can be met (indicated with a dashed arrow), the latter conditions are not evaluated as a successful match has already occurred. In case all matching changehandlers are dealt with (see Figure 5.3(b)), first all preconditions are evaluated, and subsequently, for both handlers where preconditions are met (steps 1 and 2), their actions are executed (steps 3 and 4).



(a) Matching first changehandler.



(b) Matching all changehandlers.

Figure 5.3: Immediate execution of actions specified in the first matching changehandler and in all matching changehandlers that have been constructed for a specific event, after successfully verifying their predefined preconditions.

5.4.3 Chaining Updates

After matching the changehandlers (either the first changehandler encountered, or all changehandlers), their associated update statements have to be collected and applied. This stage depends on the type of execution mechanism. In case deferred execution is applied, all update statements from the matched changehandlers have to be collected before executing them. When immediate execution is used, the statements have to be executed while inspecting them.

The earlier introduced Algorithm 5.1 defines the execution steps of the deferred execution model. In the first step, update collection, statements are collected from the matched changehandlers. Algorithm 5.5 explains how this task is performed. First, a check is done to investigate whether any changehandlers match the change event. If this is the case, the update statements from every changehandler in the set of matched changehandlers are collected. If no changehandler matches the change event, the change event itself is applied to the ontology. In lines 7-14, each update statement is treated as a change event, representing the implementation of chaining. This part is similar to Algorithm 5.1, except for the fact that updates are not applied to the ontology, because this has to happen at the end of the process when using deferred execution. In the end, the algorithm returns a list of update statements that need to be applied to the ontology.

As described in Algorithm 5.6, for immediate ontology updating, no update lists are returned. In contrast to deferred updating, the updates are immediately applied to the ontology. For immediate updating, the algorithm first checks whether any changehandler exists in the list of matched changehandlers. If this is the case, for each update statement in each of the matched changehandlers, a change event is fired as shown in Algorithm 5.2 using the update as the change event. In this way, we provide a mechanism for chaining. If no changehandler matches the change event, the change event itself is applied to the ontology.

Figure 5.4 shows a typical example of an event triggering a chained sequence of updates, while – for the sake of simplicity – assuming immediate execution of matching handlers. In the example, an event is matched to a single changehandler, of which the preconditions are met (step 1). Subsequently, the actions are either executed or (if appropriate changehandlers are available) thrown as new events in step 2. These events trigger new changehandlers to be matched (steps 3 and 4), and in turn result in more actions to be performed (step 5) and/or new events to be thrown (step 6), triggering more changehandlers (steps 7 and 8), hereby extending the chain with possibly even more changehandlers indicated with a dashed arrow.

Algorithm 5.5: Update collection from matched changehandlers (collectUpdates).

```

Task      : collect updates from a list of matched changehandlers using deferred execution
Input     : ontology  $O$  consisting of axioms,
             change event  $op(Ax)$  where  $op \in \{add, del\}$  and  $Ax$  is a set of axioms,
             list of changehandlers  $matchedHandlers$  that match the change event
Output    : list of update statements  $updateList$ 
1 // Check whether any changehandler matches the change event
2 if not  $matchedHandlers.isEmpty$  then
3   // Loop through all matched changehandlers
4   foreach  $matchedHandler$  in  $matchedHandlers$  do
5     // Loop through all update statements in the changehandler
6     foreach  $update$  in  $matchedHandler.updates$  do
7       // Chaining: add the update or the replaced update actions from other
8       //           changehandlers to the list of update statements
9       // Find changehandlers that match the update event
10       $newMatchedHandlers \leftarrow matchHandlers(O, update);$ 
11      // Collect updates from changehandlers
12       $newUpdateList \leftarrow collectUpdates(O, update, newMatchedHandlers);$ 
13      // Add the update statements to the list
14       $updateList.add(newUpdateList)$ 
15    end
16  end
17 else
18   // There is no changehandler matching the change event; therefore, the change
19   // event itself is added to the list of update statements
20    $updateList.add(op(Ax))$ 
21 end
22 return  $updateList$ 
```

Algorithm 5.6: Update application from matched changehandlers (applyUpdates).

```

Task      : apply updates from a list of matched changehandlers using immediate execution
Input     : ontology  $O$  consisting of axioms,
             change event  $op(Ax)$  where  $op \in \{add, del\}$  and  $Ax$  is a set of axioms,
             list of changehandlers  $matchedHandlers$  that match the change event
Output    : updated ontology  $O$ 
1 // Check whether any changehandler matches the change event
2 if not  $matchedHandlers.isEmpty$  then
3   // Loop through all matched changehandlers
4   foreach  $matchedHandler$  in  $matchedHandlers$  do
5     // Loop through all update statements in the changehandler
6     foreach  $update$  in  $matchedHandler.updates$  do
7       // Chaining: fire the update as an update event; this way, the update can
8       //           be handled by appropriate changehandlers
9        $updateOntology(O, update, matchedHandler);$ 
10    end
11  end
12 else
13   // There is no changehandler matching the change event; therefore, the change
14   // event itself is applied
15    $O.apply(op(Ax));$ 
16 end
17 return  $O$ 
```

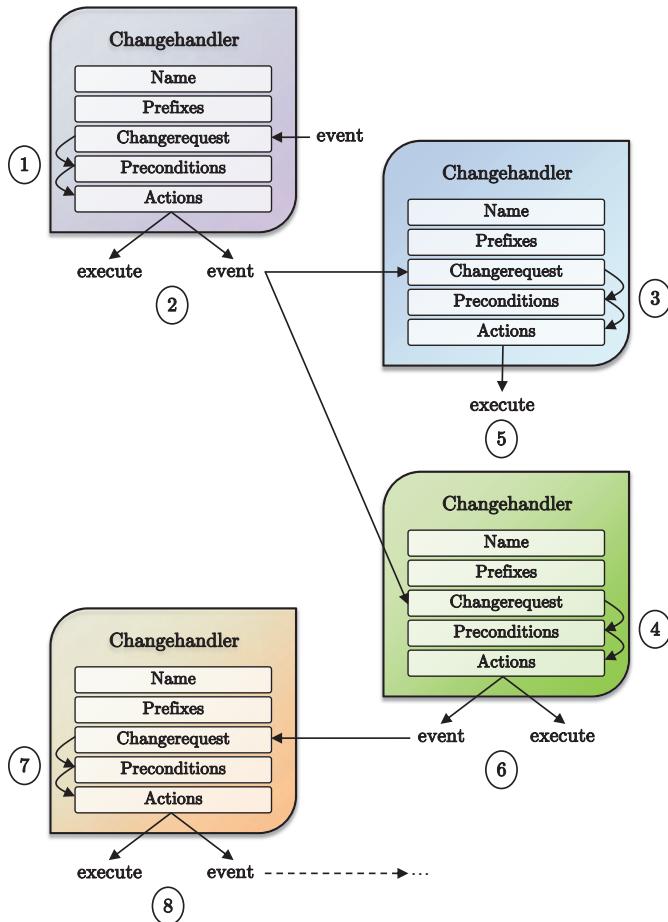


Figure 5.4: Immediate execution of actions specified in a chain of all matching changehandlers that have been constructed for a specific event, after successfully verifying the handlers' predefined preconditions.

5.4.4 Looping Updates

Applying updates to the ontology and thereby changing the ontology can trigger new changehandlers to become matched, which can be used for applying additional updates in case the event is handled more than once. Hence, we introduce the possibility to iterate over the changehandlers with the same event and apply updates until there are no matching changehandlers left. In this way, the effect of the update actions can be checked and additional updates can be applied.

Algorithm 5.7: Looped deferred ontology updating (*updateOntology*).

Task : update ontology with deferred execution of updates and looping
Input : ontology O consisting of axioms,
 change event $op(Ax)$ where $op \in \{add, del\}$ and Ax is a set of axioms
Data : changehandlers $matchedHandlers$ that match their changerequest and meet their
 precondition according to the provided change event,
 list of update actions $updateList$ to be applied to the ontology
Output : updated ontology O

```

1 // Find matched changehandlers
2 matchedHandlers ← matchHandlers( $O$ ,  $op(Ax)$ );
3 // Collect updates from changehandlers
4 updateList ← collectUpdates( $O$ ,  $op(Ax)$ , matchedHandlers);
5 // Apply updates to ontology in deferred way
6 foreach update in updateList do
7   |  $O$ .apply(update);
8 end
9 // Execute this algorithm again to check for additional updates
10 if not matchedHandlers.isEmpty then
11   | updateOntology( $O$ ,  $op(Ax)$ );
12 end
13 return  $O$ 
```

Algorithm 5.8: Looped immediate ontology updating (*updateOntology*).

Task : update ontology with immediate execution of updates and looping
Input : ontology O consisting of axioms,
 change event $op(Ax)$ where $op \in \{add, del\}$ and Ax is a set of axioms
Data : changehandlers $matchedHandlers$ that match their changerequest and meet their
 precondition according to the provided change event
Output : updated ontology O

```

1 // Find matched changehandlers
2 matchedHandlers ← matchHandlers( $O$ ,  $op(Ax)$ );
3 // Apply updates to ontology in an immediate way
4  $O \leftarrow applyUpdates(O, op(Ax), matchedHandlers);$ 
5 // Execute this algorithm again to check for additional updates
6 if not matchedHandlers.isEmpty then
7   | updateOntology( $O$ ,  $op(Ax)$ );
8 end
9 return  $O$ 
```

Ontology update looping could be implemented through the addition of a call to the *updateOntology*(...) methods of Algorithms 5.1 (deferred) and 5.2 (immediate) at the end of both algorithms, using the same change event and ontology as input. Algorithms 5.7 and 5.8 implement looping for deferred and immediate executions, respectively. Before the updated ontology is returned, the *updateOntology*(...) method is called recursively to ensure that additional updates are handled until no updates are available.

In Figure 5.5, a typical example of ontology update looping is presented, where after immediate execution of a matching changehandler in steps 1 and 2, the initial event is rethrown in step 3, after which preconditions are met that belong to another changehandler (step 4). The associated actions are subsequently executed in step 5. In the example, no subsequent iterations are needed, as no preconditions of associated changehandlers are met after the second iteration.

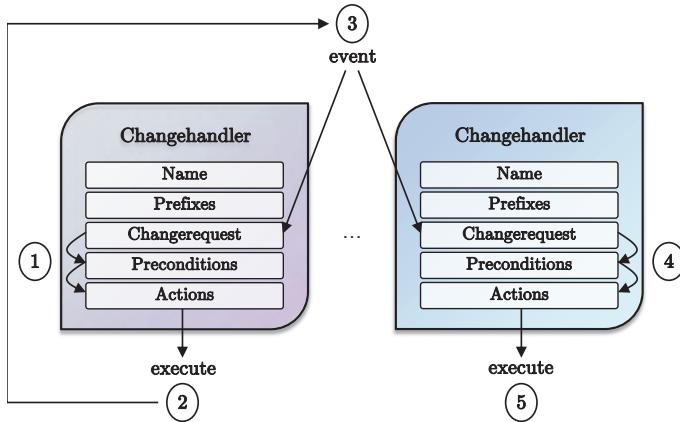


Figure 5.5: Immediate looped execution of changehandlers that have been constructed for a specific event, after successfully verifying the handlers’ predefined preconditions.

5.5 Implementation

OUL, including the proposed extensions, has been implemented as the stand-alone software package OULx, providing for event-driven ontology updates. We currently employ this package in the Hermes News Portal (Frasincar et al., 2009), a Java-based news personalization tool implementing the Hermes framework (Frasincar et al., 2009). Hermes makes use of an ontology for classifying and querying news items. The Hermes domain ontology has to be up-to-date with the latest news and hence needs automatic ontology updates. Based on the information extraction plugin for the Hermes News Portal, i.e., Aethalides, information extracted from news items can be used for updating the ontology.

A key aspect of the Hermes News Portal is its financial ontology. For its updates, the ontology is dependent on information extracted from financial news messages, e.g., product releases, CEO appointments, bankruptcies, etc. We have implemented the OULx update mechanisms and have subsequently connected them to the information extraction processes of Aethalides. The Aethalides plugin makes use of the Hermes Information Extraction Engine, which is used for matching user-created information extraction rules with text in news items.

To integrate automatic ontology updating, each new news item is processed and information (in the form of events) is extracted using the user-created rules. After validating the extracted information, the ontology is updated using OULx update rules. In our implementation, the execution of the SPARQL WHERE clauses and the SPARQL/Update statements, as well as ontology updating is performed using Jena (The Apache Software

Foundation, 2013). In order to work with the latest developments in the Semantic Web, we updated ARQ, the query engine in Jena, to version 2.8.8, which features SPARQL 1.1. The changehandlers can be loaded via a plain text file that contains changehandlers specified in the proposed syntax. Parsing and compiling of changehandlers is performed via a compiler created with JavaCC (Sun Microsystems, 2013).

5.6 Evaluation

In order to evaluate the extensions made to OUL, we analyze the characteristics of each proposed execution model. As it is difficult to perform a quantitative analysis and as there are no benchmarks available for OUL, we discuss at a qualitative level the advantages and disadvantages of each execution model. We assume all queries are chained (non-chained queries as originally proposed by OUL are also supported), which provides us with eight execution models:

- Immediate, looped execution of first matching changehandler;
- Immediate, non-looped execution of first matching changehandler;
- Immediate, looped execution of all matching changehandlers;
- Immediate, non-looped execution of all matching changehandlers;
- Deferred, looped execution of first matching changehandler;
- Deferred, non-looped execution of first matching changehandler;
- Deferred, looped execution of all matching changehandlers;
- Deferred, non-looped execution of all matching changehandlers.

Deferred execution of matching changehandlers could lead to erroneous updates, and hence it usually does not make sense to make use of the last four execution models. For example, it could be the case that several changehandlers can originally match, but after executing their corresponding updates in a deferred mode, the updates of the previous matches could be made invalid. Due to the nature of deferred execution, these updates would still be executed. On the other hand, deferred updating could possibly lead to more efficient updates in case multiple changes are to be made to the same entity, as these actions could be merged and transformed into simplified update statements. Additionally, duplicate actions can be merged, eliminating duplicate action executions. So, if

deferred updating is used, some caution is required, making sure there are no conflicting dependencies between update actions and change requests.

When comparing models that execute the first matching changehandler with those that execute all changehandlers, one could make the following observations. The latter method is computationally more intensive due to the increased complexity of the execution mechanism. Conversely, updates are more efficient, as in one pass all the matched changehandlers are dealt with, hence eliminating the need for multiple user-triggered iterations.

In case of looped execution models, the advantage is that ontology updates performed during a pass that trigger new changehandlers to be matched are taken into account, hereby improving the efficiency of the ontology updating process, as no separate runs are needed. The looped execution models are on the other hand harder for users to grasp due to the repeated event generation until no changehandler matches the event.

There is a trade-off between easiness of writing update rules and their efficient execution. The all matching and/or looped variants are more efficient due to the automatic execution and possible optimization of their complex actions, while the first matching and/or non-looped counterparts are more intuitive and thus foster easier development of update rules. Also, it should be noted that for chaining there is an increased level of automation, as users do not have to manually trigger updates resulting from earlier updates (as in case of the OUL execution model), as these are automatically handled.

5.7 Conclusions

The Ontology Update Language (OUL) is based on SQL triggers and focuses on an event-driven ontology update specification. By creating changehandlers, containing an event description, a precondition, and a list of SPARQL/Update statements, update actions are executed when events occur and preconditions are met. OUL features the creation of Event-Condition-Action rules, hereby enabling automatic updates. We identified some drawbacks at language as well as execution model levels, and proposed extensions to address these.

Syntax-wise, in order to facilitate more complex expressions, we extended OUL so that it also supports negation and prefixes. Our main contribution however lies in the extension of OUL's execution mechanism. We incorporated immediate updating, as opposed to deferred updating. Also, we added an internal triggering mechanism for changehandlers called updates chaining, allowing for automatic event triggering based on the matched changehandlers' actions. This contributes to the usability of the language by separating

atomic update actions and thus delivering modularity and an increased possibility to reuse changehandlers. Also, we added support for looping for repetitive treatment of an event. Last, it is now also possible to execute all event-related changehandlers, instead of just the first matching handler. The here proposed extensions are viable, provided that technical experts who are accustomed to the update language work together with experts of the knowledge domain.

As future work we would like to evaluate the termination of changehandlers, i.e., which conditions need to be satisfied by a set of changehandlers so that, for any incoming events, the matching changehandlers should always terminate. For this purpose we plan to reuse results from termination of rule-based updates for databases (Ray and Ray, 2001). Alternatively, one could look into developing a principled information extraction language that combines information extraction and ontology updates. For this purpose, we plan to integrate the Hermes Information Extraction Language with OUL, including the here proposed extensions.

Chapter 6

Event-Based Stock Trading Strategies[†]

In this chapter we present a framework for automatic exploitation of news in stock trading strategies. Events are extracted from news messages presented in free text without annotations. We test the introduced framework by deriving trading strategies based on technical indicators and impacts of the extracted events. The strategies take the form of rules that combine technical trading indicators with a news variable, and are revealed through the use of genetic programming. We find that the news variable is often included in the optimal trading rules, indicating the added value of news for predictive purposes and validating our proposed framework for automatically incorporating news in stock trading strategies.

[†]This chapter is based on the article “W. Nuij, V. Milea, F. Hogenboom, F. Frasincar, and U. Kaymak. An Automated Framework for Incorporating News into Stock Trading Strategies. *IEEE Transactions on Knowledge and Data Engineering*, 26(4):823–835, 2014.”

6.1 Introduction

Financial markets are driven by information. An important source of information is news communicated by different media agencies through a variety of channels. With the increasing number of information sources, resulting in high volumes of news, manual processing of the knowledge being conveyed becomes a highly difficult task. Additionally, given that this information is time-sensitive, especially in the context of financial markets, selecting and processing all the relevant information in a decision-making process, such as the decision whether to buy, hold, or sell an asset is an especially challenging task. This environment motivates a need for automation in the processing of information, to the extent that investment decisions where the news factor plays an important role can be based on an automatically generated recommendation that takes into account all news messages relevant to a certain financial asset.

In previous work we have devised lexico-semantic patterns for information extraction from news that extend the well-known lexico-syntactic patterns with semantic aspects (Borsje et al., 2010; IJntema et al., 2012). Using information extracted from text in a financial context recently enjoys increasing attention. Das and Chen (2007) extract investor sentiment from stock message boards. The prediction of bankruptcy of firms, as well as fraud, based on textual data from the Management Discussion and Analysis Sections (MD&A) of 10-K reports is investigated by Cecchini et al. (2010). A popular Wall Street Journal column is used for investigating asset prices and trading volumes by Tetlock (2007), and in later work, Tetlock (2008) uses financial news stories for the prediction of stock returns and firms' future cash flows. Thus, the qualitative data may emerge from different sources, and can be used for the prediction of different financial aspects of firms' performance.

We focus on information presented in textual format, i.e., financial news messages with a particular focus on companies listed under the FTSE350 stock index. The research question addressed is how the information communicated through textual news messages can be automatically incorporated into trading strategies. We use a three-step approach consisting of: (i) extracting the relevant events, as well as the involved entities, from the text of the news messages, (ii) associating an impact with each of the extracted events, and (iii) making use of the impact of news events in trading strategies.

Upon extracting the events and associating these with a predefined impact, trading rules based on news can be derived. We only consider technical trading indicators as part of these trading rules, but the approach can be easily extended to incorporate other indicators, e.g., those initiating from fundamental analysis. Technical trading has been

used previously for financial forecasting (Leigh et al., 2002; Mehta and Bhattacharyya, 2004), thus motivating our choice for this approach. The constructed trading strategies are expressed as trees, where the leaves are technical indicators or news event indicators and internal nodes represent the conjunctive and disjunctive logical operators. These trading strategies generate a buy or sell signal for the assets they are applied to, and are determined through genetic programming where a pool of possible trading strategies is tested on historical stock data.

We hypothesize that, if the proposed framework is valid, news will be included in the trading strategies generated through genetic programming. Additionally, the trading strategies that we derive in this way should generate positive returns. The first hypothesis comes from the idea that, when providing a genetic program with a pool of variables without the restriction that all these variables should be included in a trading strategy, only the variables that are maximizing the returns will be selected. Trading strategies including a news variable will thus indicate that the content of the news messages has been quantified in a way that enables generation of profit beyond the ability of trading rules based solely on technical analysis. The second hypothesis states that, next to generating trading strategies based on news, the resulting rules should also be able to obtain a positive return.

Although the main purpose of this chapter is not to find the best trading rules by using news, the presented results can be applied to (trading) algorithms that are used in daily practice. For this purpose, additional aspects need to be taken into account, as for example the transaction costs and the rule creation speed. Existing trading algorithms (many proprietary) could benefit from our approach by additionally employing the news signals as shown in our proposed framework. Because our framework is customizable by means of its various parameters, it can be used for other stock markets than the ones considered here, providing for a general methodology of including the news component in a trading algorithm.

The remainder of this chapter is structured as follows. First, we present previous work on the relationship between news and the stock market, and the type of events that are proven to influence stock prices in Section 6.2. Next, we provide an initial, quantitative investigation of the relationship between news messages and the stock market in Section 6.3. Subsequently, we discuss the technical indicators that we use for deriving stock trading strategies in Section 6.4, after which we introduce our framework for automated trading based on news and discuss the results of validating the framework in Sections 6.5 and 6.6, respectively. Last, we give some practical considerations and conclude this chapter in Sections 6.7 and 6.8, respectively.

6.2 Related Work

Three aspects can be considered regarding the relationship between news and the stock market: (i) there is evidence that a relationship exists between news announcements and financial markets, (ii) the impact of events on financial markets can be quantified, and a list of relevant events can be identified, and (iii) the relationship between information in the form of news and financial markets is not a trivial one. One aspect left aside in this chapter relates to mining news messages for assessing market response, which has been extensively surveyed by Mittermayer and Knolmayer (2006).

Mitchell and Mulherin (1994) investigate the relation between the number of news announcements and trading activity. The research is focused on whether the amount of information that is publicly reported affects the trading activity and the price movements in the stock market. Here, information is defined and quantified by the number of daily announcements released by the Dow Jones & Company newswire. All news messages are assigned equal importance, regardless of the type of event being described. The results indicate a statistically significant positive correlation between the number of daily news announcements and trading activity. If the number of announcements increases with 100%, the trading activity will increase with 38%. The relation becomes stronger only if those news messages are selected that, besides being published through the wire service, are published in a newspaper the next morning.

The relation between news announcements and monthly returns is also investigated by Chan (2003). Several stocks are selected with at least one news story in a certain month. The news messages are divided into ‘news winners’ (price increased after announcement) and ‘news losers’ (price decreased after announcement). The abnormal returns are measured for 36 months after the month when the news was published. The results are compared to a group ‘no news’ containing those companies which had no news in a certain month. The authors conclude that stocks exhibit abnormal returns after public news.

The effect of analyst recommendations, with a focus on buy advices, is studied by Kim et al. (1997). First, the authors test whether the advice issued especially for clients (before the opening of the stock market) contains information and then perform the same test but following the official release of the advice (to the main public). The authors find a strong relation between an initial coverage with a buy recommendation and a reaction in the stock market.

The relation between earnings announcements and trading volume around the announcement date is investigated by Ewalds et al. (2000), focusing on AEX exchange stocks

from 1994 to 1999. A significant positive increase in trading volume is found around an announcement. The increase in trading activity is the largest at the announcement date. The robustness of the relation is checked for small and large companies. Both categories have a significant relation with trading activity, but the relation is much stronger regarding small companies compared to large companies. A possible explanation is that there is less information available about small companies. Another relation was found between the date of the announcement and trading volume. The longer a company waits with revealing the earnings, the smaller the change in trading volume. A possible explanation is that the expectations are more accurate in that case, i.e., analysts have more time and information (earnings from competitors) to accurately predict earnings.

Up until now, different meanings have been assigned to the word news when studying the relation to the stock market. The employed news sources are arbitrary, they contain different news messages, and are not complete. Although a relation is apparent, it is necessary to zoom into real-life events and quantify the relation between these events and returns. The list of events that possibly affect the stock price is extremely large, but in the remaining paragraphs we focus on a limited number hereof, considered to be of increased relevance in financial markets.

A management change event is a change in the set of individuals holding the title Chief Executive Officer (CEO), president, or chairman of the board (Warner et al., 1988). The reaction after a management change indicates whether the market considers this event as important. A stock return after a management change contains:

- The information effect (negative): the management performance is worse than expected by the market.
- The real effect (positive): the change is in shareholders' interest. If a company performs very bad, a management change could mean a new vision, strategy, etc., so the expectations about the companies' future results could be revised. The news is received positively.

The authors did not find a general relation between a management change and abnormal stock returns. Only on the day of the announcement a statistically significant price movement was noticed, but the direction could be both positive and negative.

The effect of a management change is studied by Bonnier and Bruner (1989). The identified average excess return from the day before until the day of an announcement is 2.479% (positive significant). Also, the title power, the company size, and the manager type have a positive, significant impact. Generally, a management change conveys bad

news about the company's performance, but a management change is received positively if the company performance is bad.

Keown and Pinkerton (1981) study the trading activity and price movements before, on, and after the day of a merger announcement. In 79% of the acquired firms a significant increase in trading volume is found one week before the announcement, compared to 3 months before that date. Approximately half of the reactions occur before the official public announcement – they start one month before the merger. The strongest reaction in 1 day is on the announcement day itself: the market reacts immediately.

The price momentum following a merger announcement is investigated by Rosen (2006). Price momentum relates to an initial market response to a merger announcement and its propagation through time, i.e., if the initial reaction is positive, it tends to continue to be so. The results indicate that if a company is associated with successful mergers in the past, this will positively influence price momentum.

Ikenberry and Ramnath (2002) study stock splits and their effect on the price, and use an NYSE sample from 1988 until 1997 with over 3000 stock splits. The authors find a 9% positive difference of abnormal returns between the split stocks and a control group, a year after the split.

The price reaction after dividend initiations and omissions is investigated by (Michaely et al., 1995) for short term (3 days) and long term (several years) using a buy and hold strategy to measure returns. In the 3 days around an initiation announcement a significant excess return of 3.4% is found. In the year before, the excess return is 15.1%. Companies with a dividend omission perform very poor in the year before the announcement, apparent from an excess return of -31.8%. Around the announcement, an excess return of -3.1% is identified. These trends continue also in the next 1 year and the next 3 years after the announcement.

These findings come to support our assumption that events that can be identified in news messages have a significant impact on stock prices and trading volumes. For this reason, we consider it worthwhile to employ such events in our analysis. In the final part of this section we focus on different properties of (public) information in the context of financial markets.

Zhang (2006) explore the degree of uncertainty of information, while hypothesizing that greater information uncertainty will lead to higher expected stock returns after good news on the one hand, and lower expected returns after bad news on the other hand. This implication is based on results from behavioural finance studies, i.e., psychological biases such as overconfidence are increased when there is more uncertainty. Here, good and bad news are defined as upward and downward analyst forecast revisions, respectively.

Evidence is found that the market reaction directly after an announcement is incomplete, i.e., bad news implies relatively lower future returns and good news predicts higher future returns.

The influence of certain forms of rumours on trading activity on the stock market is evaluated by van Bommel (2003) through a dynamic model with three kinds of rumours: honest, bluffing, and cheating. It is concluded that spreading rumours makes economic sense. Rumours can increase stock demand and drive the price above the real price. In case of cheating with false rumours, followers will not use the trader's rumours, causing the rumourmonger to lose his reputation.

Our approach does not focus on representing and reasoning with complex knowledge contained in news messages, but rather focuses on single events. It would be possible to design an ontology of events in the Web Ontology Language (OWL) (Bechhofer et al., 2004) if the context is static, although a temporal web ontology language such as tOWL (Frasincar et al., 2010; Milea et al., 2008, 2012a,b) should be more suited for representing and reasoning with the facts exposed by the news messages. However, as we focus on single events extracted from news, we do not rely on an approach based on ontologies.

Similar work regarding the extraction of optimal trading rules based on technical indicators related to price is presented by Allen and Karjalainen (1999). However, unlike the current research, news are not used in trading strategies. Also, in previous work (Hogenboom et al., 2012b,c, 2013c) we have successfully used news events for financial risk analysis by improving the historical Value at Risk method.

Our current approach is novel in that it does not focus on a particular event type, but rather on a thesaurus of events that play a significant role in financial markets. Rather than focusing on news volume, we extract relevant events from market announcements and try to include them in trading rules. For this, we employ genetic programming that can choose between different variables in creating profitable trading rules. The variables originate in technical analysis, except for the news-related variable.

6.3 News and Stock Markets

Our analysis of the relationship between news and the stock market, as apparent from the collected data set is focused on discovering the influence that news have on the share price of the concerned companies, as well as on whether this influence can be captured through the extraction of events from news messages and employing a predefined impact for determining the direction of this influence on prices.

6.3.1 Event Information Extraction

The event information extraction from the news messages is based on recognizing a pre-defined set of events as well as the affiliated entities. For this we rely on the ViewerPro tool (available at <http://www.semlab.nl/portfolio-item/viewerpro-semantic-text-analysis/>), a proprietary application able to extract events from text-based data.

ViewerPro is an application created by SemLab that enables the identification of events in news messages. These events can be used to determine the impact of a news item on an equity. ViewerPro turns enormous amounts of unstructured news into structured trading information. Once the unstructured news information is fed in the ViewerPro system, it undergoes several (proprietary) processing steps in order to filter out unwanted information and select solely that which is relevant. Applied procedures are (amongst others) metadata filtering, parsing, gazetteering, stemming, and automatic pattern matching.

The ViewerPro system relies on a domain specific knowledge repository, i.e., an ontology with properties and lexical representations of financial entities (companies). First, concepts from the domain ontology are matched in incoming news items. Subsequently, using a proprietary heuristic based on semantical, morphological, syntactical, and typographical inputs, the list of concepts is segmented into groups of related concepts. Last, ViewerPro identifies events (predefined semantic concepts describing important message content) by means of pattern matching.

Large amounts of news messages are filtered for equity-specific news and the semantic analysis system of ViewerPro interprets the impact of every individual news message.

6.3.2 Descriptive Statistics of the Data Set

The data set we employ consists of a database of historical company share prices as well as a collection of news messages related to these companies. The company data set consists of all firms included in the FTSE350 stock index at August 1st, 2008. Stock prices are scraped from Yahoo! Finance ticker data. The news data set is collected through the Reuters news feed, and concerns all 350 companies listed under FTSE350. Both data sets cover the period January 1st, 2007 until April 30th, 2007. The news data set provides a set of 5,157 events. However, only a subset hereof is employed for our study. The selection of these relevant events is based on three criteria:

- News articles issued on days when the stock exchange is closed are not considered;
- Duplicate events are removed;
- Rare events (< 0.5% of all events) are omitted.

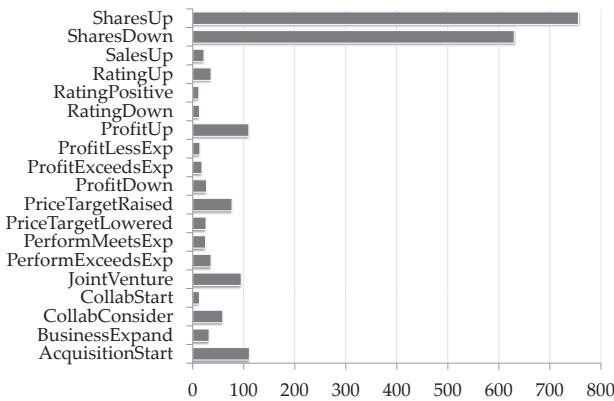


Figure 6.1: Frequency of events in the data set.

We do not include articles issued on days when the stock exchange is closed as the events contained in these messages will not have an immediately quantifiable impact on the stock price. Since several events can occur during the period when the stock exchange is closed, associating these events with changes in price over this period will introduce additional variance with regard to which event precisely influences the change in price.

At times, news messages may be repeated to provide updates on an event described in a previous message. This results in events that are on the same day, concern the same company, and are identical to another event previously extracted on the same day. Since these news messages describe the same events, it suffices to only consider them once and thus incorporate the associated impact for the event in the stock price projection only once.

Infrequently occurring events, i.e., events occurring in less than 0.5% of the news messages, are removed from the data set as considering them would negatively influence the statistical validity of our conclusions. Moreover, the impact of such isolated events is difficult to assess with confidence.

Upon considering these four criteria on the event data set we have collected, the original sample of 5,157 events is reduced to 2,112. An overview of these events, as well as their frequencies in the event data set, is presented in Figure 6.1.

6.3.3 News and Share Prices

The impact of news on stock prices is assessed using relative returns, based on end-of-day data, i.e., closing prices P . For a single asset, a return is computed as:

$$r_i = \frac{P_{i+n} - P_i}{P_i} \times 100 , \quad (6.1)$$

where i represents the day before the event and n represents the number of days over which the return is calculated, with $n > 0$.

In case multiple events of the same type appear in different days, regarding the same asset, the return is averaged for the number of days, as follows:

$$R_i = \frac{\sum_{j=1}^N r_j}{N} , \quad (6.2)$$

where N is the number of days where events of this type occurred.

To correct the returns for the general market sentiment, we focus on excess returns. The excess return is calculated as the individual return of an asset that is achieved in excess of the market return, i.e., the return of the main index in which the asset is included:

$$a_i = r_i - r_i^I , \quad (6.3)$$

where r_i^I denotes the return of the index employed as benchmark.

When dealing with multiple events of a certain type appearing in different days and with excess returns, we correct these returns for the number of days:

$$A_i = \frac{\sum_{j=1}^N a_j}{N} , \quad (6.4)$$

where N is the number of days where events of this type occurred.

For the results presented in this section, the benchmark index used to compute excess returns is the FTSE350 index. An overview of the results is presented in Tables 6.1 and 6.2. For each event we compute the absolute and excess returns for the day of the event, R_0 and A_0 , as well as the returns following the event one, two, five, and ten days after the event is made public. For each of the events, we compute the percentage of events for which the direction of the return corresponds to the sign of the impact, i.e., positive returns in the case of positive impacts and negative returns in the case of negative impacts, and we denote this by d . Additionally, we compute the two-tailed t-test significance for the returns obtained for each of the events, and report this as p . The impacts reported in Tables 6.1 and 6.2 have been manually determined by finance experts from Semlab. Last, three remarks are in place when interpreting these results:

- Multiple events may determine asset prices, and thus the returns, while not all these events could be captured through the news messages used for the analysis. However,

we assume that the largest share of the reported returns is captured by the reported events.

- Reactions to events, in terms of price changes, may initiate before the event is public. By relying on the asset's closing price on the day previous to the event, we incorporate most of the anticipation preceding an event.
- When manually assessing an event's impact on stock prices (reported under the impact column in Tables 6.1 and 6.2) the assumption is made that no other interactions, involving, for example, other events, have a significant influence on the price.

An initial inspection of Tables 6.1 and 6.2 reveals that in nearly 90% of the event types, the direction of the R_0 returns corresponds with the sign of the impact assessed by experts. The two events where this is not the case are the *collaboration consideration* and *performance meets expectations* events. However, the expert impact associated with these events is only slightly positive, while the generated returns are slightly negative. Thus, based on the small number of events on which this impact is assessed, the assumptions listed previously, and the small difference between the expert impact and the generated returns, we consider the impact assigned by experts to be trustworthy in the absence of additional data. Last, the slightly negative returns are not significant at the 95% level.

When considering R_0 , the event that generates the highest return is the *shares up* event, producing an average of 1.63%, backed up by the fact that 85% of this type of events generated a positive return. Presumably, not all events in this category generate a positive return due to the fact that, in some occasions, this event co-occurs with another event that generates a decrease in price that dominates the increase associated with the *shares up* event.

In the short run, i.e., when considering the R_0 , R_1 , and R_2 returns, we find three events for which the returns are both statistically significant, as well as showing the same direction as the impact determined by experts, in all three cases: *shares up*, *shares down*, and *rating up*. In the long run, however, we find more events for which both returns are statistically significant as well as being correctly captured by the manually determined impacts: *shares up*, *rating up*, *rating positive*, *profit up*, and *shares down*. From this we conclude that, for most events, the impacts are observable at longer time intervals after the event is reported.

When excess returns are considered, the short run exhibits four events for which the returns are significant at the 95% level and the direction of the return corresponds with

Event	Impact	Freq.	R_0	d	p	R_1	d	p	R_2	d	p	R_5	d	p	R_{10}	d	p
SharesUp	2	756	1.63	85	0.00	1.65	80	0.00	1.53	73	0.00	1.74	72	0.00	2.27	69	0.00
RatingUp	2	36	1.55	83	0.00	1.82	72	0.00	1.89	75	0.00	2.52	69	0.00	2.31	72	0.01
CollabStart	2	13	1.06	46	0.32	1.73	62	0.18	1.95	62	0.14	1.95	77	0.10	1.30	62	0.20
RatingPositive	1	12	0.84	75	0.25	0.96	58	0.32	1.51	75	0.28	3.23	75	0.02	4.26	92	0.02
ProfitExceedsExp	3	18	0.74	50	0.39	1.99	61	0.11	1.72	61	0.19	2.87	56	0.08	2.61	61	0.12
AcquisitionStart	3	111	0.40	62	0.08	0.56	59	0.06	0.53	55	0.12	0.90	59	0.02	0.90	62	0.06
SalesUp	2	22	0.33	64	0.44	0.56	55	0.37	0.35	55	0.65	1.17	64	0.29	1.44	59	0.23
PriceTargetRaised	2	77	0.24	51	0.31	0.54	56	0.13	0.65	58	0.11	0.83	56	0.08	2.35	71	0.00
BusinessExpand	1	32	0.21	53	0.70	0.83	56	0.16	0.69	56	0.34	1.52	66	0.08	2.13	72	0.02
JointVenture	1	95	0.18	53	0.31	0.12	47	0.59	0.10	52	0.72	0.49	56	0.14	0.78	53	0.05
PerformExceedsExp	3	36	0.11	58	0.82	0.35	50	0.52	0.25	47	0.70	1.49	53	0.11	0.58	56	0.53
ProfitUp	2	110	0.08	50	0.80	0.22	50	0.53	0.26	52	0.54	1.25	56	0.03	1.73	64	0.01
CollabConsider	1	59	-0.05	49	0.81	-0.08	54	0.83	-0.17	49	0.69	0.19	59	0.67	0.11	59	0.85
PerformMeetsExp	1	25	-0.21	40	0.66	-0.11	48	0.79	0.31	56	0.57	0.55	64	0.46	0.75	56	0.42
ProfitDown	-2	27	-0.94	48	0.21	-0.52	52	0.49	0.05	48	0.95	0.33	52	0.78	0.83	52	0.56
RatingDown	-2	13	-0.96	54	0.15	-0.98	62	0.14	-1.32	69	0.11	-0.65	54	0.54	0.04	46	0.98
PriceTargetLowered	-2	26	-1.17	73	0.16	-1.44	77	0.12	-1.77	73	0.07	-1.50	65	0.14	-1.51	50	0.20
SharesDown	-2	630	-1.38	81	0.00	-1.49	71	0.00	-1.44	69	0.00	-1.20	62	0.00	-0.91	59	0.00
ProfitLessExp	-3	14	-2.52	64	0.06	-2.33	71	0.04	-2.04	71	0.08	-1.38	57	0.40	-1.29	64	0.35
		2,112		60		60		61		61		62		62		62	

Table 6.1: Average returns R_x for different time intervals of x days after a financial event (*Event*), alongside the event's associated predefined predicting factor for future price movement (*Impact*), frequency (*Freq.*), percentage of returns d which went into the right direction (positive if impact is positive), and two-tailed t-test significance value p at the 95% level.

Event	Impact	Freq.	A_0	d	p	A_1	d	p	A_2	d	p	A_5	d	p	A_{10}	d	p
RatingUp	2	36	1.49	81	0.00	1.88	72	0.00	1.89	72	0.00	2.46	69	0.00	2.02	69	0.03
SharesUp	2	756	1.32	80	0.00	1.31	74	0.00	1.26	68	0.00	1.59	68	0.00	1.58	64	0.00
CollabStart	2	13	1.02	46	0.26	1.62	69	0.15	1.76	62	0.12	1.74	77	0.10	0.83	62	0.23
ProfitExceedsExp	3	18	0.98	67	0.24	1.90	61	0.13	1.52	61	0.23	2.38	67	0.12	1.48	50	0.35
RatingPositive	1	12	0.83	67	0.26	1.21	67	0.17	1.69	75	0.11	2.88	75	0.03	3.35	67	0.03
AcquisitionStart	3	111	0.44	60	0.03	0.59	56	0.03	0.59	52	0.05	0.61	50	0.10	0.49	49	0.25
PerformExceedsExp	3	36	0.37	53	0.33	0.46	61	0.29	0.44	53	0.38	1.15	53	0.17	0.22	47	0.79
PriceTargetRaised	2	77	0.30	52	0.15	0.69	53	0.04	0.85	58	0.02	0.85	49	0.05	1.70	61	0.00
ProfitUp	2	110	0.19	46	0.48	0.43	55	0.15	0.45	55	0.17	1.21	55	0.01	1.44	56	0.01
SalesUp	2	22	0.16	59	0.69	0.39	59	0.47	0.26	55	0.72	1.04	55	0.27	0.94	59	0.35
BusinessExpand	1	32	0.15	38	0.74	0.66	56	0.23	0.46	50	0.46	1.17	53	0.10	1.11	53	0.18
Joint Venture	1	95	0.14	49	0.36	0.18	54	0.39	0.13	51	0.58	0.13	49	0.65	-0.02	44	0.96
CollabConside	1	59	-0.14	44	0.48	-0.10	53	0.75	-0.20	49	0.57	-0.24	58	0.49	-0.63	41	0.16
PerformMeetsExp	1	25	-0.21	40	0.64	-0.32	40	0.47	-0.01	44	0.99	-0.27	48	0.67	-0.58	52	0.51
PriceTargetLowered	-2	26	-0.94	73	0.24	-1.36	85	0.16	-1.59	73	0.12	-2.24	73	0.04	-2.34	58	0.07
RatingDown	-2	13	-1.11	69	0.08	-0.91	69	0.13	-1.10	69	0.20	-0.40	46	0.71	-1.06	54	0.42
ProfitDown	-2	27	-1.14	63	0.14	-0.75	52	0.31	-0.19	52	0.82	0.22	63	0.83	0.12	63	0.93
SharesDown	-2	630	-1.18	80	0.00	-1.30	73	0.00	-1.30	73	0.00	-1.49	68	0.00	-1.46	64	0.00
ProfitLessExp	-3	14	-2.42	64	0.09	-2.42	71	0.04	-2.31	79	0.05	-1.36	71	0.30	-2.19	79	0.12
		2,112	59		62		61		61		60		60		57		

Table 6.2: Abnormal returns A_x for different time intervals of x days after a financial event (*Event*), alongside the event's associated predefined predicting factor for future price movement (*Impact*), frequency (*Freq.*), percentage of returns d which went into the right direction (positive if impact is positive), and two-tailed t-test significance value p at the 95% level.

R_x	r	p	A_x	r	p
R_0	0.844	0.000	A_0	0.878	0.000
R_1	0.851	0.000	A_1	0.863	0.000
R_2	0.804	0.000	A_2	0.827	0.000
R_5	0.789	0.000	A_5	0.753	0.000
R_{10}	0.662	0.002	A_{10}	0.712	0.001

Table 6.3: Pearson's correlation r between impact and returns R_x , and between impact and abnormal returns A_x for different time intervals of x days after a financial event, together with their corresponding two-tailed t-test significance values p at the 99% level.

the sign of the impact: *rating up*, *shares up*, *acquisition start*, and *shares down*. For the long run, the same is found for the events: *rating up*, *shares up*, *rating positive*, *price target raised*, *profit up*, and *shares down*.

The linear relationship between the predefined event impact and the generated absolute returns is quantified by means of Person's correlation. For all time intervals in Table 6.3, we find a significant, positive correlation between the returns and impacts.

Additionally, we report the values for Pearson's correlation for the predefined impact and abnormal returns. Again, for all time horizons, we find strong and significant positive correlations. In the case of excess returns, the values for Pearson's correlation are higher than in the case of absolute returns, indicating that correcting for the index leads to results that are more in line with the expectations.

Based on the results, two conclusions can be drawn. First, the events that are selected and extracted from news messages can be employed in trading strategies, as in most cases these events provide the ability to generate positive returns. Second, the predefined impact associated with the extracted events is a good reflection of the impact of these events on stock prices, as apparent from the Pearson correlation test presented in this section.

6.4 Technical Trading

This section focuses on the technical trading indicators used in trading strategies generated through genetic programming. The indicators included in the study are: the simple moving average (SMA), the Bollinger band (BB), the exponential moving average (EMA), the rate of change (RoC), momentum (MOM), and moving average convergence divergence (MACD). The choice for these indicators is based on their widespread use in technical trading (Achelis, 2000).

6.4.1 Simple Moving Average

The SMA averages the last 20 days of the price of a stock (Achelis, 2000), and is computed as:

$$M_i = \frac{\sum_{i=1}^N P_i}{N}, \quad (6.5)$$

where P_i represents the price on day i . The average is calculated over a fixed period of 20 days prior to the day for which the average is calculated, i.e., $N = 20$, which is standard for this indicator. A buy signal is generated when the price crosses the moving average in an upward movement, while a sell signal is generated when the price crosses the moving average in a downward movement.

6.4.2 Bollinger Bands

The Bollinger band is a technical indicator which creates two ‘bands’ around a moving average (Achelis, 2000). These bands are based on the standard deviation of the price. It is assumed that the price will move within these bands, around the moving average. If the volatility is high, the bands are wide and when there is little volatility the bands are narrow. The lower and upper Bollinger bands can be calculated as:

$$L = M - 2 \times \sigma_M, \quad (6.6)$$

$$U = M + 2 \times \sigma_M, \quad (6.7)$$

where σ_M stands for the volatility of moving average M . A buy signal is generated when the price is below the lower band, which is regarded as an oversold situation. A sell signal is generated at an overbought situation, when the price is above the upper band.

6.4.3 Exponential Moving Average

The exponential moving average (EMA) aims to identify trends by using a short and a long term average (Achelis, 2000). When the averages cross each other, it is the start of a new trend. The short term average is set at 5 days and the long term average at 20 days:

$$E_i = \frac{2}{N+1} \times (P_i - E_{i-1}) + E_{i-1}, \quad (6.8)$$

where P_i represents the price on day i , and N is the number of days. The initial EMA is calculated using the SMA, in our case for 5 and 20 days respectively starting from the first observation, as previously described. When the short term average crosses the long

term average upwards, a buy signal is generated. A sell signal is generated when the short term average crosses the long term average downwards.

6.4.4 Rate of Change

The rate of change (RoC) is an indicator that calculates the difference between the closing price P_i of the current day i and the closing price P_{i-10} of 10 days earlier (Achelis, 2000), according to the following equation:

$$C_i = \frac{P_i - P_{i-10}}{P_{i-10}} . \quad (6.9)$$

If the RoC starts decreasing above 0 (a peak was reached), a sell signal is generated. If it starts increasing below 0, a buy signal is generated.

6.4.5 Momentum

The momentum indicator uses exactly the same formula as the RoC. Instead of creating a buy signal after a peak, it creates a buy signal when the momentum crosses the 0 level upwards (Achelis, 2000). A sell signal is generated when the RoC crosses the 0 level downwards.

6.4.6 Moving Average Convergence Divergence

The moving average convergence divergence (MACD) is a technical indicator that subtracts two exponential averages from each other, namely the 12 and the 26 day exponential average (Achelis, 2000). The mathematical formula for the MACD is:

$$D_i = E[12]_i - E[26]_i . \quad (6.10)$$

A buy signal is generated when the MACD reaches the 0 level in an upward motion. A sell signal is generated when the MACD breaks through the 0 level in a downward motion.

6.4.7 Performance of Technical Trading Indicators

We now analyze the performance of the individual technical indicators when considered separately from any other indicators. In Table 6.4, we present the returns generated by each technical indicator over different time intervals, both for buy and sell signals. The frequency shows how many signals are generated by the indicator. The returns show the

Indicator	Freq.	Buy Signals				
		R_0	R_1	R_2	R_5	R_{10}
SMA(20)	1,663	1.927	2.069	2.197	2.463	3.512
BB	2,014	-1.608	-1.374	-1.170	0.110	0.180
EMA(5,20)	870	1.717	1.860	1.859	2.122	3.350
RoC(10)	4,387	-0.417	-0.290	-0.245	-0.053	0.245
MOM	1,988	1.499	1.444	1.637	1.938	2.759
MACD(12,26)	667	1.310	1.324	1.309	1.568	2.882

Indicator	Freq.	Sell Signals				
		R_0	R_1	R_2	R_5	R_{10}
SMA(20)	1,737	-1.888	-1.932	-1.923	-1.496	-0.805
BB	2,971	1.563	1.530	1.499	1.595	1.827
EMA(5,20)	922	-1.662	-1.654	-1.586	-0.933	-0.488
RoC(10)	2,937	0.723	0.637	0.953	1.564	2.370
MOM	2,049	-1.310	-1.151	-1.248	-1.021	-0.629
MACD(12,26)	581	-1.276	-1.106	-1.162	-0.106	0.317

Table 6.4: Returns R_x for different time intervals of x days after a financial event, resulting from signals generated by technical indicators (*Indicator*), of which the frequencies (*Freq.*) are also displayed.

average return surrounding a buy or sell signal in the given time frame, e.g., R_{10} represents the 10 day return. The best performing technical indicator for buy signals is the simple moving average, followed by the exponential moving average. Positive returns generated through buy signals are displayed in bold, as these are desirable. Table 6.4 additionally displays the performance of the technical indicators when sell signals are considered. Note that negative returns in this case are desirable (and hence printed bold), since higher magnitudes of a negative return indicate better performance of the sell signal (you would lose less money if you sell). Again, the two best performing technical indicators are the simple moving average and the exponential moving average.

6.5 A News-Based Trading Framework

Next, we introduce a framework for incorporating news in stock trading strategies. The framework assumes that events have been extracted from news messages and are available together with the date on which the events took place. Additionally, a predefined impact should be assigned to each event, allowing the news variable to be included in the trading strategies. For deriving the optimal trading strategies, we rely on genetic programming.

Genetic programming (Koza, 1992) is a technique where the potential solutions are represented as computer programs rather than numerical values encoded in some manner. Starting from a (usually randomly generated) initial population, genetic programs attempt to improve the fitness of individuals over successive generations through a process inspired by natural evolution. During this process, individuals are altered, usually based on their fitness values, by combining them with other individuals (crossover), or by slightly modifying some parts of the individual with a predefined probability (mutation). In this chapter, genetic programming is used for finding optimal trading strategies based on technical indicators and news. Genetic programming has previously been used in the design of decision support systems, e.g., by Zhao (2007) and Fan et al. (2006).

The trading strategies we determine take the form of trees that, when evaluated, return a Boolean value: true, when a trading signal is generated, or false, when no signal is generated and thus no action has to be taken. The trading strategies include at least one technical indicator or a news variable. Most often, the trading strategies include multiple variables, that may be either technical indicators or the news variable, connected by the logical operators ‘and’ and ‘or’. An example of a trading strategy that may be generated is given in Figure 6.2. This rule generates a trading signal when the simple moving average generates a trading signal simultaneously with at least one of the exponential moving average and rate of change indicators. The fitness of a trading strategy is computed based on the return that it generates on the data set that we use. The returns are computed as indicated in Section 6.3.

The employed genetic programming algorithm for determining the optimal trading strategies is presented in Algorithm 6.1. We start from a random initial population of trees, and generate new populations of trading strategies by applying crossover and mutation on the population from the previous iteration. Crossover consists of selecting two trading strategies, and determining two random crossover points, i.e., one for each tree.

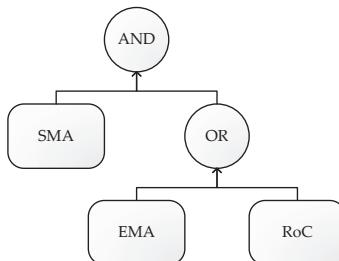


Figure 6.2: Example of a typical trading rule.

Algorithm 6.1: Genetic programming approach
for determining optimal trading strategies.

```

Require :  $\alpha \geq 0$ : minimum improvement
           $\beta > 0$ : maximum times of no improvement
           $\gamma > 0$ : population size
           $0 < \rho \leq \gamma$ : number of parents
           $0 \leq \mu \leq 1$ : mutation probability

1  $\pi = \text{generateRandomPopulation}(\gamma);$ 
2  $\sigma_{old} = -\infty;$ 
3  $\sigma_{new} = \text{calcFitness}(\pi);$ 
4  $b = 0;$ 
5 while  $b < \beta$  do
6    $\text{addIndividual}(\pi', \text{getBest}(\pi, \sigma_{new}));$ 
7   while  $|\pi'| < |\pi|$  do
8      $\theta = \text{selectRandomParents}(\pi, \sigma_{new}, \rho);$ 
9      $\vartheta = \text{crossOver}(\theta);$ 
10     $\vartheta' = \text{mutate}(\vartheta, \mu);$ 
11     $\text{addIndividual}(\pi', \vartheta')$ 
12  end
13   $\pi = \pi';$ 
14   $\sigma_{old} = \sigma_{new};$ 
15   $\sigma_{new} = \text{calcFitness}(\pi);$ 
16  if  $\sigma_{new} - \sigma_{old} \leq \alpha$  then
17    |  $b = b + 1;$ 
18  end
19  else if  $b > 0$  then
20    |  $b = 0;$ 
21  end
22 end
23 return  $\pi$ 

```

Next, the subtrees generated under the crossover point are exchanged between the two trading strategies, thus resulting in two new rules that are added to the new population. Mutation only relates to the technical indicators included in a trading strategy, and consists of a slight change in the parameters of the randomly selected technical indicator, e.g., changing the number of days used by the simple moving average from 5 to 7. The stopping condition for the algorithm relates to the improvement in the best solution found, i.e., when the optimal solution cannot be improved in a number of generations, the algorithm stops.

We summarize the proposed framework in Figure 6.3. As illustrated in the figure, the events are extracted from the news messages represented in free text format, and constitute the input to the algorithm. The historical price data constitutes an individual input to the search algorithm, used for computing the performance of the trading strategies, but is simultaneously used to derive the values for technical trading indicators, another input to the algorithm. Finally, the optimal trading strategies are determined through genetic programming.

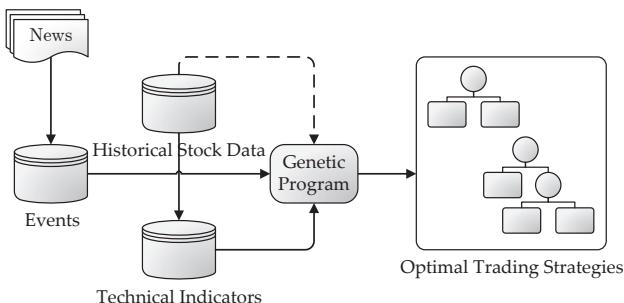


Figure 6.3: News-based trading framework.

6.6 Experiments and Results

In this section we provide an overview of the validation of our proposed framework for including news in stock trading strategies. First, we focus on the performance of the news variable taken individually, and then in combination with each of the technical indicators we consider. We subsequently present the performance of optimal trading rules as determined through genetic programming, and discuss these results.

6.6.1 Performance of Individual Events

When trading strategies are built only by using the news variable, we generate the returns displayed in Table 6.5. Here, a buy signal is generated when events are encountered that are known to produce an R_0 of at least 0.5%, as shown in Tables 6.1 and 6.2. Similarly, a sell signal is generated when the R_0 is below -0.5%. In Table 6.5 we present the results for buy and sell signals individually, for different time horizons.

A comparison with the results obtained by using trading strategies based only on technical indicators, as presented in Table 6.4, reveals that the news variable performs well. In the case of buy signals, the trading strategies based solely on the news variable are consistently outperformed by the simple moving average and the exponential moving average, but only slightly. In case of sell signals, the situation is similar, although the exponential moving average is not consistently outperforming the trading strategies based on the news variable.

The good overall performance obtained by using technical indicators and news, respectively, suggests that combining these indicators could result in better trading strategies. In the next section, we look at how each of the technical indicators performs when considered in combination with the news variable.

Signal	Freq.	R_0	R_1	R_2	R_5	R_{10}
Buy	818	1.563	1.625	1.531	1.808	2.301
Sell	579	-1.578	-1.640	-1.606	-1.332	-1.061

Table 6.5: Returns R_x for different time intervals of x days after a financial event, resulting from signals generated by news (*Signal*), of which the frequencies (*Freq.*) are also displayed.

6.6.2 News and Technical Indicators

Next, we consider the performance of the individual technical indicators when trading strategies combine each of them with the news variable. Again, buy and sell signals are considered separately, and the returns are presented for different time horizons.

Table 6.6 presents the returns generated with buy signals when news and technical indicators are considered together. Positive returns generated through buy signals are displayed in bold, as these are considered desirable results. When news and technical indicators are considered together, the generated returns are higher than when these variables are considered individually. Out of the six combinations, four consistently generate positive returns at all time horizons. The highest observed return is generated by the combination of news and moving average convergence divergence, for time horizon R_{10} .

Buy Signals						
Indicator	Freq.	R_0	R_1	R_2	R_5	R_{10}
News & SMA(20)	108	3.005	2.888	2.748	3.236	3.954
News & BB	18	-0.841	-0.587	0.011	0.374	-0.174
News & EMA(5,20)	54	3.284	3.081	3.013	3.433	4.246
News & RoC(10)	68	-0.098	0.036	-0.194	-0.060	0.369
News & MOM	100	3.058	2.857	2.976	3.339	3.874
News & MACD(12,26)	33	3.566	3.610	3.563	4.040	5.371

Sell Signals						
Indicator	Freq.	R_0	R_1	R_2	R_5	R_{10}
News & SMA(20)	93	-3.177	-3.700	-3.849	-3.353	-3.166
News & BB	19	0.702	0.445	0.485	0.311	-1.163
News & EMA(5,20)	42	-3.790	-3.894	-3.977	-3.529	-3.654
News & RoC(10)	28	0.211	-0.414	0.231	0.319	0.196
News & MOM	78	-3.304	-3.748	-3.969	-3.417	-3.369
News & MACD(12,26)	31	-3.796	-4.217	-4.307	-4.555	-3.240

Table 6.6: Returns R_x for different time intervals of x days after a financial event, resulting from signals generated by news and technical indicators (*Indicator*), of which the frequencies (*Freq.*) are also displayed.

In Table 6.6 we additionally present the returns generated through sell signals when the individual technical indicators are considered together with news. Again, results are presented for different time horizons. The overall conclusion is that the combination of the two indicators generally outperforms the trading strategies based on the indicators taken separately. From the six combinations, four consistently generate desirable returns at all time horizons. Again, the highest return is achieved through the combination of news and moving average convergence divergence, but this time at time horizon R_5 .

These results allow us to conclude that combining individual technical indicators with the news variable for determining trading strategies enables higher returns than when technical indicators and the news variable are considered separately. We next move on to generating more complex trading strategies, that rely on multiple technical indicators, possibly in combination with the news variable, for generating trading signals.

6.6.3 Optimal Trading Strategies

The optimal trading strategies are determined through genetic programming, as outlined earlier. The initial population employed by the genetic program consists of 50 randomly generated trading rules. For our experiments, we let the algorithm run for 15 generations with a mutation rate of 0.5. For each new generation, all parents are selectable for crossover. Experiments are performed on both buy and sell signals, and take into consideration a selection of 2,000 data points from our data set. Initial experiments have shown that these settings generate optimal results, while minimizing processing time. Strategies are generated for holding stocks for 1, 3, and 5 days, and are evaluated on the remaining 40,957 data points.

In Table 6.7 we show the results obtained when the fitness of the generated trading rules is computed as the relative return after a number of days from the generation of a signal. The table displays the results of trading strategies making use of buy signals and sell signals. All but one of the best performing buy rules when considering a one-day period include news as a relevant variable, with generated returns of 2.17% to 2.39% on the training set, and 2.33% to 3.34% on the test set. The simple moving average is included in all the rules, confirming the performance this indicator achieved when trading rules were considered that take into account only the individual technical indicators. When observing the top five trading strategies and their generated returns three days after the generation of a buy signal, we again note that the simple moving average is included in many trading strategies. Also, the news variable is part of the best performing trading rule, generating a return of 2.64% (2.96%). However, when the time horizon consists of

		Buy Signals			
		Train		Test	
x	Tree	Freq.	R_x	Freq.	R_x
1	SMA(27) & News	21	2.388	74	3.338
	SMA(27) & (SMA(15) News)	34	2.225	617	2.470
	SMA(27) & (SMA(17) News)	41	2.177	666	2.415
	SMA(20) & MOM(5)	33	2.201	549	2.368
	(SMA(27) & SMA(21)) (SMA(15) News)	48	2.166	818	2.334
3	MOM(4) & News	34	2.636	126	2.958
	MOM(4) & SMA(27)	30	2.345	397	2.594
	SMA(15) & SMA(16)	86	1.889	1583	2.074
	SMA(17) SMA(20)	120	1.627	2068	1.841
	News	196	1.220	611	1.665
5	SMA(19) & News	28	2.246	86	2.927
	SMA(18) & News	29	2.045	90	2.856
	SMA(22) & SMA(27)	50	2.081	858	2.500
	SMA(24) & (MOM(6) News)	34	1.919	484	2.705
	MOM(3) EMA(21,83)	180	1.570	3334	1.480

		Sell Signals			
		Train		Test	
x	Tree	Freq.	R_x	Freq.	R_x
1	SMA(27) & News	23	-2.233	59	-3.357
	MOM(7) & News	28	-2.048	81	-3.087
	MOM(7) & SMA(28)	13	-2.315	351	-2.614
	MOM(7) & MOM(4)	29	-2.084	647	-2.182
	MOM(7) & MOM(3)	31	-2.333	642	-2.145
3	SMA(24) & News	20	-2.578	60	-3.747
	(MOM(4) & News) & MOM(5)	36	-2.464	105	-2.949
	MOM(4) & News	36	-2.464	105	-2.949
	SMA(24) & (MOM(5) & News)	24	-2.088	428	-2.451
	MOM(7) & MOM(5)	43	-2.131	704	-2.174
5	MOM(7) & News	28	-2.598	81	-3.283
	MOM(7) & (SMA(24) SMA(21))	22	-2.965	427	-2.628
	MOM(7) & SMA(16)	38	-2.783	739	-2.194
	(SMA(23) News) & (SMA(24) BB)	214	-1.368	1776	-1.751
	News	147	-1.369	416	-1.619

Table 6.7: Optimal strategies (*Tree*), their frequencies (*Freq.*), and returns (R_x) if stocks are held x days (FTSE350 data set).

three days, the news variable is included less often in the optimal trading strategies, while momentum is of greater importance than before. For five-day rules, again, the simple moving average is included in many well performing trading strategies. News is included

in the optimal trading strategy, as well as in two others, generating returns between 1.92% and 2.25% on the training set, and 2.71% to 2.93% on the test set. Also, momentum and the exponential moving average can be found in the best trading strategies.

The table additionally shows the resulting top trading rules using the returns of sell signals and suggests that, as is the case when considering buy signals, news is an important indicator both on the short and on the long run, although for sell signals, news tends to be slightly less important for one-day sell strategies and more important for three- and five-day strategies. When observing one-day sell strategies, news is only included in the two best performing rules, whereas strategies for holding stocks a longer period include news more frequently. Average returns are comparable to the ones from the buy signals as well, yet it should be noted that momentum plays a more important role as a technical indicator for sell rules than for buy rules, and the simple moving average is used less frequently.

In order to validate the results, we perform the same experiments on another data set, concerning all 500 companies listed under S&P500 at September 1st, 2012, covering the period between May 1st, 2010 and September 30th, 2010. News is collected through the Reuters news feed and stocks are scraped from Yahoo! Finance ticker data. The set has similar characteristics as the default data set that is used throughout this chapter, although there are more types of events, albeit with a less frequent occurrence. Pruning is performed in the same way as done with our default data set, resulting in 2,308 events after removing simultaneous and duplicate events, rare events, and events occurring on non-trading days.

Table 6.8 presents an overview of the generated trading rules based on this data set. All parameter settings for the genetic programming algorithm have remained unchanged. Again, a training set of 2,000 items is used for rule generation, resulting in a test set of 50,121 remaining data points for evaluation. For both buy and sell signals, the rules presented in the table confirm that news is often included in trading rules and hence is an important factor in these rules. A more detailed analysis of the FTSE350 and S&P500 rules is presented below.

When analyzing the generated rules for buy signals, we observe that in each of the best performing rule groups, news is included as a signal. For each of the evaluated horizons, the generated trading rules for S&P500 buy signals show many resemblances with those for the FTSE350 data set in terms of the technical indicators used. In both data sets, many rules include news and/or make use of the simple moving average. Also, the number of times news is included in the best performing trading rules is approximately the same

		Buy Signals			
x	Tree	Train		Test	
		Freq.	R_x	Freq.	R_x
1	SMA(27)	94	2.599	2420	2.900
	SMA(18) News	142	2.236	3365	2.707
	SMA(18) MACD(12,26)	150	2.190	3700	2.629
	News (SMA(26) News)	115	2.328	2839	2.607
	SMA(26) (MOM(5) News)	264	2.095	6683	2.186
3	MOM(4) & SMA(15)	51	3.349	1268	3.343
	SMA(20)	117	2.669	2823	2.904
	EMA(22,105) SMA(28)	98	2.743	2610	2.747
	SMA(28) News	108	2.711	2762	2.705
	(News (SMA(22) SMA(18))) SMA(20)	127	2.671	3035	2.694
5	News SMA(23)	123	2.877	2970	2.785
	SMA(25) SMA(15)	197	2.613	4767	2.780
	SMA(22) News	127	2.806	3035	2.778
	(SMA(20) News) News	133	2.624	3200	2.715
	(EMA(18,105) News) SMA(27)	118	2.582	3029	2.616

		Sell Signals			
x	Tree	Train		Test	
		Freq.	R_x	Freq.	R_x
1	SMA(22) & MOM(5)	43	-2.965	1003	-2.962
	SMA(22) & SMA(14)	48	-3.118	1346	-2.920
	SMA(28) & SMA(22)	51	-3.095	1360	-2.822
	News (SMA(24) EMA(24,94))	132	-2.285	3049	-2.203
	(EMA(24,94) SMA(23)) News	133	-2.316	3111	-2.162
3	SMA(16) EMA(16,136)	126	-2.469	3388	-2.352
	EMA(25,145) SMA(16)	125	-2.406	3361	-2.337
	SMA(16) News	136	-2.231	3473	-2.263
	MOM(4) SMA(16)	305	-1.891	7674	-1.708
	News MOM(3)	282	-1.612	6638	-1.488
5	EMA(14,137) SMA(25)	106	-2.375	2698	-2.382
	SMA(15) News	144	-2.042	3597	-2.317
	SMA(25) SMA(15)	183	-2.434	4630	-2.294
	News SMA(27)	117	-1.931	2725	-2.148
	EMA(19,111) MOM(5)	212	-1.621	5079	-1.523

Table 6.8: Optimal strategies (*Tree*), their frequencies (*Freq.*), and returns (R_x) if stocks are held x days (S&P500 data set).

for both data sets for each of the evaluated horizons. The returns on the other hand are slightly higher for the S&P500 data set.

For sell signals, the role of news in trading rules for S&P500 equities is visible, yet it is less prominent. For each of the evaluated horizons, news is included in the best performing trading rules, yet not in the top rules. This could be explained by the fact that the frequency of events that trigger sell signals in this data set when compared to our default data set is lower. The latter observation underlines a key issue. When there is enough breaking news, information can play a crucial role in trading. However, when more news is published on less significant events, it is a less reliable trading indicator. Moreover, the signal scarcity is reflected in the operands used in the generated trading rules. While on our FTSE350 data set, often the ‘&’ operand is used for news inputs, on the S&P500 data set the ‘|’ operand is more frequently used. This implies that news is more often a strengthening signal for the technical indicators in the FTSE350 data set compared to the S&P500 data set. Another observation that can be made is that the momentum technical indicator is used less often, in favor of the simple moving average indicator, causing the compositional differences between rules for buy and sell signals to be smaller now. Furthermore, generated returns for the various resulting trading rules are different from the ones created for the FTSE350 data set, which is caused by the fact that the S&P500 data set covers different equities and a different time span.

6.7 Practical Considerations

When applying the proposed techniques to a new domain, one should consider the various steps of transforming news into events, converting events to signals, and generating rules based on signals. Of crucial importance are having at one’s disposal a domain expert who is able to define concepts and (impacts of) events, a small training set for determining signals, and computing power for learning trading rules. Subsequently, the learned rules can be deployed in real-time applications for the considered domain.

Transforming news into signals for trading algorithms requires the extraction of features and events from text. Our proposed natural language processing approach, which makes use of part-of-speech tagging, lemmatizing, and – most importantly – ontology concept identification, can be employed for extracting domain-specific events from text. Our ViewerPro-based implementation is flexible in that the concepts and events of interest can be specified by the user. Hence, extending the news processing and event recognition approach to other domains can be achieved with minimal additional effort. In case of new concepts or events, expert knowledge can be employed for defining concept specifications and event impacts. Subsequently, the event-associated returns can be computed

based on a training set in order to determine event signals that comprise the input of our subsequent trading rule learning algorithm.

In practice, the results of this study can be directly applied to trading algorithms by generating news-driven signals, indicating whether to buy, sell, or hold a specific stock. These signals can be generated as proposed in this chapter. Alternatively, more complex signals can be obtained, which can be based not only on news, but also on other (numerical) inputs. When events extracted from news are turned into trading signals, one is able to devise trading rules that make use of various signals, not only stemming from news, but also originating from various other inputs such as moving averages. As we successfully empirically demonstrated for two domains related to FTSE350 and S&P500 companies, by following our genetic programming approach, one could learn well-performing trading rules for a specific domain and for a range of horizons by constructing a set of rules that use signals from news or other numerical inputs, and by iteratively mutating and crossovering the best performing rules (i.e., the ones that generate the highest returns) in a subsequent processing step. To create good trading rules for a specific domain or scenario, the algorithm parameters such as mutation and crossover rates need to be adjusted to the scenario at hand, by means of the procedure explained below.

In order to apply our approach on a new market (e.g., NASDAQ100 companies), one should learn parameters based on a representative training set for the specific domain. For this, simple learning approaches such as a hill climbing procedure or more advanced approaches like genetic algorithms or other meta-heuristics can be used. In all cases, optimal parameter values are determined while maximizing returns. Additionally, if performance is a limitation, execution time should also be minimized. Parameters to be optimized are the initial population size (i.e., the number of rules), the number of generations (iterations), the mutation rate (i.e., the fraction or percentage of rules that are mutated), and crossover (i.e., the percentage of parent rules that are eligible for creating new offspring).

6.8 Conclusions

We presented a framework for incorporating news into stock trading strategies. The trading strategies that we consider may include (in addition to the news variable) any number of technical trading indicators. The news variable is quantified based on the events extracted from the text of news messages and the assignment of an expert-defined impact to each of these events. Our results indicate that the assigned impacts correlate well with the returns generated by these events when tested against real data based on FTSE350 equities.

The selected technical indicators are also tested, and the individual performance of each indicator is reported. Additionally, combinations of individual technical indicators and the news variable are investigated. The results indicate that adding the news variable to each of the indicators generates higher returns than when each of the variables is considered alone. This suggests that considering the news variable indeed can lead to higher returns, thus making it worthwhile to employ trading rules that, next to technical indicators, make use of the events that are relevant for a certain company.

Last, a genetic program is used to discover complex trading rules based on technical indicators and news-based signals. For this purpose, we consider three time horizons when computing the fitness of the trading strategies based on the generated returns, namely one, three, and five days after the generation of a buy or sell signal. We conclude that, in many cases, news is a relevant variable for trading rules, and its inclusion in trading strategies leads to higher returns than when this variable is not considered. Experiments on a contrastive data set containing data on S&P500 equities confirm the previous observations, and hence underline the importance of taking into account news as an additional input for trading rules. Based on the positive returns of the generated rules, we also conclude that the proposed framework is appropriate for including news in technical trading strategies.

Our results indicate that the inclusion of news into stock trading strategies can be achieved by extracting the events from the text of the news messages and associating an impact with these events (based on stock price variations for an event). This impact can later be used in the derivation of optimal trading strategies, where the news variable, consisting of the predefined impact, is used next to technical indicators. Returning to the two hypotheses stated in the introduction, namely that news will be included in the optimal trading strategies if news is a relevant variable and that these trading rules should generate positive returns, we conclude that the news variable has been quantified in a meaningful way, confirming our first hypothesis that news would be included in the optimal trading strategies. Additionally, all trading strategies that include news events generate a positive return, thus confirming our second hypothesis.

Future work will focus on including more indicators, technical or non-technical in nature, in the variable pool from which trading strategies are generated. Additionally, a more fine-grained analysis of the news messages, e.g., identification of event-related information such as the involved actors, should provide more information for generating trading strategies. Last, considering the interaction between events occurring within the same day, or within finer-grained time intervals, will provide a deeper understanding of the way that news impact stock prices and may lead to more profitable trading strategies.

Chapter 7

Event-Based Risk Analysis[‡]

VALUE at Risk (VaR) is a tool widely used in financial applications to assess portfolio risk. The historical stock return data used in calculating VaR may be sensitive to rare news events that occur during the sampled period and cause trend disruptions. Therefore, we examine whether VaR accuracy can be improved by considering rare news events, identified using a Poisson distribution, as an additional input in the calculation. Our experiments demonstrate that VaR predictions for rare event occurrences can be improved by removing the event-generated disturbance from the stock prices for a small, optimized time window.

[‡]This chapter is based on the article “F. Hogenboom, M. de Winter, F. Frasincar, and U. Kaymak. A News Event-Driven Approach for the Historical Value at Risk Method. *Expert Systems with Applications*, 2013. To Appear.”

7.1 Introduction

Over the years, Value at Risk (VaR) has become a widely adopted risk measure in the financial world and is now also a requirement for regulatory purposes despite its acknowledged limitations in terms of interpretability and mathematical properties (Artzner et al., 1999; Rockafellar and Uryasev, 2002). Such alternatives as the expected shortfall or conditional VaR (CVaR) (Rockafellar and Uryasev, 2000; Street, 2010), which measure the market risk of a portfolio and are more sensitive to the tail of the loss distribution than conventional VaR, have better properties but have not yet become standards. VaR is typically used in the field of finance to quantify the risk of loss on a portfolio of financial equities, and it is defined as a threshold value such that the probability that the loss on the portfolio over a given time horizon does not exceed a certain value at a given confidence level (Olson and Wu, 2010).

Lately, VaR has been criticized for its vulnerability in times of financial crisis (Asche et al., 2013). Research has shown that VaR estimation for emerging markets is difficult during periods of financial turmoil, as the forecasts of most models tend to be overly conservative (Dimitrakopoulos et al., 2010). The estimates quickly become inaccurate, particularly if asset prices are highly correlated, which is the case in the oil and gas industry, among others (Asche et al., 2013). For developed markets, these effects appear to be less of an issue. From these recent discussions, we deduce that practitioners who apply VaR for risk calculations generally assume that there are no unexpected trend breaks in portfolio prices. In reality, we are regularly faced with deviations from trends that are mainly caused by emerging events, which are reported in news messages. Emerging events can be related not only to crisis situations, such as announcements of losses or even bankruptcies, but also to times of prosperity, such as profit announcements and acquisitions. All of these events have the potential to greatly impact today's financial markets, as they disrupt trends – positively or negatively – and can thus cause traders to react, seeking opportunities or minimizing daily losses.

According to the weak form of the efficient market hypothesis, news that contains information on an equity is not perfectly incorporated in the price when it is published. Studies have reported on the existence of such a delay (Fama, 1965), caused by initial over or underreactions to the news. Additionally, news events affect the volatility of equities (Mitchell and Mulherin, 1994). The usage of information extracted from text in a financial context has proven to be a vital strategy in many financial applications (Chan, 2003; Ikenberry and Ramnath, 2002). Thus, considering news events in VaR calculations (which are based on returns distributions) could be beneficial, as volatility is the standard

deviation of the distribution of returns (Byström, 2009; Engelberg and Parsons, 2009; Goonatilake and Herath, 2007; Kalev et al., 2004). It would be useful for traders to react to these news events in a timely and accurate fashion before the competition and incorporate an additional news input into the mostly numerical high-frequency trading algorithms used today. One could account for these destabilizing effects in equity stock prices in other calculations that use prices as input, such as VaR predictions.

In this chapter, we hypothesize that we can improve VaR computations by introducing financial news events (Borsje et al., 2010; Hogenboom et al., 2011a, 2013b) as an additional input. In our experiments, we use the proprietary ViewerPro (Semlab, 2013) software for the extraction of 2010 and 2011 ticker data and news events for different equities. Using a Poisson distribution, we identify the irregular events. Subsequently, we clean the ticker data of rare event-generated noise and obtain a data set that is a more accurate representation of the expected returns distribution. We also seek to optimize the time window for which the cleaning is conducted by evaluating the accuracy of the calculated VaR for different configurations. Although the time span covered by the data set is associated with a financial crisis, this fact should have no effect on our experiments. As stated earlier, for highly correlated stock prices, estimates may become inaccurate during times of economic crisis; however, in our data set, we sought to have a set of companies at different stages in their economic life cycles, circumventing such negative effects. Moreover, although many events can potentially be discovered that could disrupt VaR predictions, this is also the case in times of economic expansion. The proposed methods are defined irrespective of the state of the economy and merely depend on emerging events, independent of the economic situation for which the data are collected.

This chapter is a continuation of previous and ongoing efforts to improve VaR calculations (Hogenboom et al., 2012b,c). Hogenboom et al. (2012c) used a fixed time window to consider the effects of an event on stock prices. Additionally, all events in the data set were considered to have potentially destabilizing effects on stock prices, whereas in (Hogenboom et al., 2012b), a Poisson distribution was introduced to distinguish rare events from frequently occurring events that are not likely to influence stock prices due to their regularity over time. Our current endeavors expand on this previous work by providing additional details on the proposed extensions to the VaR calculation method.

The remainder of the chapter is organized as follows. First, we describe approaches related to this research in Section 7.2. Next, we describe our proposed method for considering news events in historical VaR calculations in Section 7.3. In Section 7.4, we present an evaluation of the proposed method. Finally, in Section 7.5, we present our conclusions and identify directions for future work.

7.2 Related Work

VaR has been widely studied as a measure of the risk of loss on a specific portfolio of financial assets, represented as a single number (Holton, 2003). VaR has become widely used in practice by corporate treasurers and fund managers. It is also used by regulators to determine the capital that financial institutions are required to have to cover their risks (Hull, 2011). According to its classical definition, for a given portfolio, probability, and time horizon, VaR can be formally described as a threshold value such that the probability of a VaR break, i.e., the loss on the portfolio over a given time horizon, exceeds this threshold values at a given probability level (Jorion, 2006). For this computation, it is assumed that there is a normal market and that no trading takes place in the portfolio.

For VaR calculations, one can distinguish among three main methods. First, the parametric method assumes a specific distribution of equity returns. Commonly employed distributions for the parametric method are the normal distribution and the log-normal distribution, which offer simplicity while maintaining robustness. However, in practice, equity returns are almost never normally distributed (Andersen et al., 2001). Second, the Monte Carlo simulation-based method predicts future returns by fitting a distribution based on historical data. In contrast to the parametric method, Monte Carlo simulation does not assume a normal distribution because it randomly samples the historical data multiple times to approximate its distribution. However, this random sampling renders the method computationally intensive, and thus, real-time application is difficult to achieve. The last method, i.e., the historical method, is the most popular method for calculating VaR because of its real-time applicability even though the computed values for some applications could contain little information on future volatilities (Pérignon and Smith, 2010). In this chapter, we employ the historical method for VaR prediction, which we extend to consider information on news events.

A recent example of a parametric method for VaR calculation is the work of Huang and Lee (2013). The authors seek to predict future daily returns by considering high-frequency 5 minute data and demonstrate the merits of using such high-frequency intra-day data over using low-frequency data alone. Their proposed method involves a single parameterized model, which is constructed based on averaged or merged high-frequency data. Thus, the model is constructed over multiple forecasts that are generated using multiple lower-frequency (daily) data sets constructed from higher-frequency (intra-day) data. In their work, three data merging methods are evaluated, i.e., combining forecast, subsample averaging, and bootstrap averaging. Although this work is similar to ours in that high-frequency data are used as input, the work differs from our current endeavors

in that we seek to predict returns within minutes or hours, whereas daily returns are predicted by Huang and Lee (2013). Moreover, the authors do not consider news events to smooth out irregularities in the high-frequency data. Finally, our approach does not rely on an underlying parametric model but rather on historical data alone.

Another example of a parametric approach is given by Bormetti et al. (2012). The authors propose a Bayesian methodology for VaR computation that employs parametric production partition models. Such models rely on clustering structures and allow one to identify anomalies in the data under the assumption that the data are normally distributed. The approach is evaluated by comparing it to maximum likelihood-based approaches. The overall performance is the same, but the authors claim that the proposed methodology provides richer information about the clustering structure and outliers. The drawbacks of the proposed approach are related to scalability and dimensionality issues resulting from an increased number of assets and parameters.

The semi-parametric approach proposed by Mancini and Trojani (2011) does not assume a normal distribution but estimates predictive distributions of (parametric) GARCH models. Evaluation of the method through a Monte Carlo simulation and empirical application illustrates that the method provides more accurate VaR forecasts than classical methods like the historical method, particularly for longer horizons of several days and in the case of outliers. An important difference between that approach and our proposed approach is that Mancini and Trojani employ a semi-parametric approach with a Monte Carlo simulation, whereas we employ a non-parametric method, the historical method.

GARCH models have also been popular in attempts to improve historical VaR calculations. For instance, Hull and White (1998) improve the VaR calculation by updating the volatility in the historical method using GARCH/EWMA models to reflect the difference between the volatility at the time of the observation and the current volatility. An important difference between the work of Hull and White and ours is that we only observe portfolios with a fixed composition (i.e., fixed weight factors) rather than regular multi-equity portfolios. We seek to avoid having interdependencies between variances (heteroscedasticity), which is often the case when observing portfolios containing various financial equities. Hull and White propose a method to update the volatility during an appropriate time interval so that the volatility becomes a more dynamic factor in the VaR calculation. Based on the mean absolute percentage error (MAPE), their work is compared to another method that involves the assignment of weights to more recent observations (Boudoukh et al., 1998). The method proposed by Hull and White (1998) appears to outperform the traditional historical method and the method proposed by Boudoukh et al. (1998) for exchange rates, but the results are mixed for stock indices.

The existence of a strong relationship between the stock market and news events has been acknowledged in many previous studies (Byström, 2009; Engelberg and Parsons, 2009; Goonatilake and Herath, 2007). Additionally, there is a proven correlation between number of news events and trading activity (Mitchell and Mulherin, 1994). However, many of the existing VaR calculation methods still do not consider these events. In practice, news information is not always fully and immediately processed in the value of shares (Rosenberg et al., 1985), and thus, for traders, reacting to news and estimating the portfolios' VaR in a timely and accurate manner is of utmost importance.

Antweiler and Frank (2006) note that those studies that do consider events often do not have extensive robustness checks. Typically, these studies employ inter-day data and are limited to a time horizon of several days before and after an event. Moreover, very little evidence is provided about what occurred in the evaluated time period. In contrast, long-term event studies are often subject to publication bias and typically focus on a single news event type. Antweiler and Frank employ a naive Bayes algorithm to classify news while employing various time windows and demonstrate that there is an initial under or overreaction to news events, followed by directly opposite stock price movements. The authors claim that news events with a mainly negative connotation, e.g., merge announcements, equity repurchases, or declining earnings, appear to yield large abnormal returns. As with most comparable studies, a serious drawback is the use of inter-day data. In contrast, in our research, we employ data that is more fine-grained, i.e., we analyze intra-day data at hourly and 5 minute intervals.

Another method developed to improve technical trading algorithms by using information extracted from news is discussed by Zhai et al. (2007). The authors use a simple text classification algorithm with a supervised learning method. However, they also integrate general market news in combination with technical indicators instead of only microeconomic news events, which is the focus of our work. Based on a real-life market simulation, the authors conclude that technical indicators and news events alone are inaccurate as estimators but that the combination of the two could possibly yield better results.

7.3 Event-Based Historical Value at Risk

To assess whether the incorporation of news into the calculation of VaR of equities improves the overall quality of the prediction, we propose the framework depicted in Figure 7.1. The framework is based on two inputs, namely, an unmodified complete list of historical stock prices, $prices_{hist}$, and a list of financial events, $events$. These inputs are extracted from several feeds using the ViewerPro application, a proprietary application

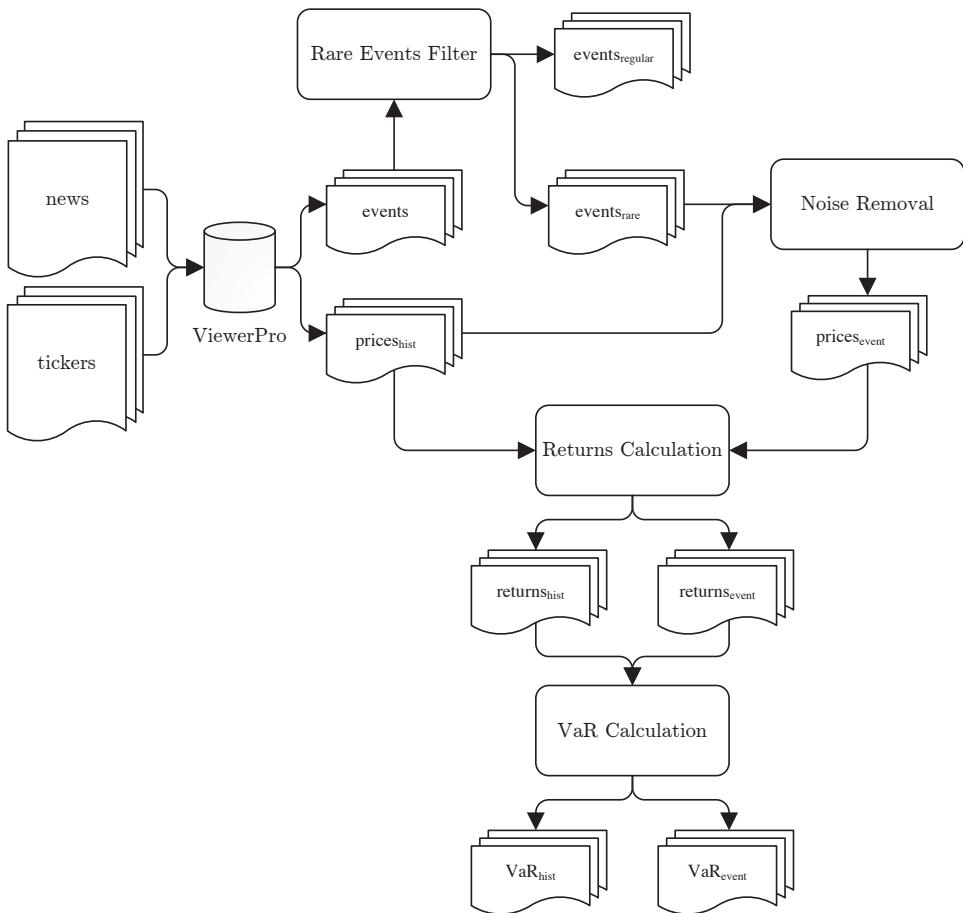


Figure 7.1: Overview of data flows and processing steps.

of SemLab, used to extract various types of events from text-based data, such as news messages. These events, some examples of which are given in Table 7.1, can be used to determine the impact of a news item on an equity.

Before using the collected equity prices $prices_{hist}$, a cleaning procedure is followed, by which prices that are recorded within stock markets' opening times are retained for further computations. In addition, to decrease the computational complexity, the time intervals between individual prices are defined per hour. News is parsed for events (stored in the list $events$) using ViewerPro's computational linguistics, semantic analysis, and formal logic procedures, which determine the positive and negative impacts of the information described in the news on the equities.

Type	Sub-types
CEO	hiring, resignation
Acquisition	consideration, start, completion, stop
Bid	receival, consideration, acceptance, drop, raise
Profit	down, up
Legal conflict	loss, resolution, win
Bankruptcy	-

Table 7.1: Examples of news event types identified by the ViewerPro software.

ViewerPro converts large quantities of data about unstructured news items into structured trading information. After feeding unstructured news items into the ViewerPro system, several proprietary processing steps are executed to filter out unwanted information. The main procedures are metadata filtering, parsing, gazetteering, stemming, and automatic pattern matching. The ViewerPro system relies on a domain-specific knowledge repository, i.e., a domain ontology with properties and lexical representations of financial entities (companies). First, concepts from the domain ontology are matched to news items. Second, the list of concepts is segmented into groups of related concepts using heuristics based on semantic, morphological, syntactical, and typographical data. Finally, the application identifies events using pattern matching with the previously extracted information.

Additionally, we identify the irregularly occurring event types from our event set, as these events are not likely to occur again and thus cause significant noise in stock rates. Poisson distributions are used in many fields to model the number of events that occur in a certain time interval, and therefore, we apply a Poisson distribution F to a test set *test*:

$$F(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad (7.1)$$

where x and λ represent the measured and expected number of occurrences in the test set *test*. For a threshold α of 0.05, for $x = 0$ (no occurrence), $F(x; \lambda) < \alpha$ for $\lambda \geq 3$, as depicted in Figure 7.2. For a training set *train*, the expected number of occurrences λ' is obtained by scaling λ by the proportion of the set cardinalities, i.e., $\lambda' = \lambda \times \vartheta$, with $\vartheta = |\text{train}|/|\text{test}|$. Therefore, in this paper we consider event types that occur $\geq 3 \times \vartheta$ in the training set as regular events, and events occurring $< 3 \times \vartheta$ as rare events. Rare events are stored in *events_{rare}* and are used in further processing steps, whereas regular events in *events_{regular}* are discarded.

Next, noise removal is performed using the identified rare events in set *events_{rare}*, which are associated with the times at which they occurred and their respective stock

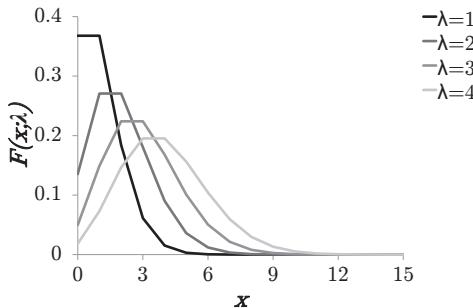


Figure 7.2: Poisson distributions for measured and expected occurrences x and λ .

rates. We adjust the collected prices $prices_{hist}$ for a time window to account for the generated noise by changing their values to the previously measured value, resulting in a list of event-corrected prices $prices_{event}$. This process is illustrated in Algorithm 7.1, in which a list of chronologically ordered hourly recorded historical prices $prices$ (containing all prices in $prices_{hist}$ for a specific stock) is processed into a set of prices $prices_{event}$ that is cleaned from the effects of an emerging event from event list events (containing the rare events from $events_{rare}$). For each historical stock price $price$ in $prices$, the algorithm checks for event occurrences by comparing the stock price time with the time of each rare event $event$ stored in events. When an event occurrence is identified, we set $impact$ to the window size $window$, which causes the value of the subsequent $price$ items to be set to the current value (we assume that no two events occur simultaneously). The value of $impact$ is decreased by one every next $price$ in price list $prices$, so subsequent price values are updated until the window size has been reached. In the case of overlapping events, the $impact$ counter is reset to the window size $window$.

Algorithm 7.1: Stock price cleaning based on news events.

Require : $prices$: array of historical stock prices and associated times
 $events$: array of rare events and associated times
 $window$: integer representing time window

```

1  $previousprice.value = prices[1].value;$ 
2 foreach  $price$  in  $prices$  do
3   foreach  $event$  in  $events$  do
4     if  $impact > 0$  then
5        $impact = impact - 1;$ 
6        $price.value = previousprice.value;$ 
7     end
8     if  $price.time = event.time$  then
9        $impact = window;$ 
10    end
11  end
12   $previousprice.value = price.value;$ 
13 end
```

Both sets of original (“*hist*”) and cleaned (“*event*”) prices are converted to sets with hourly returns. We compute the return set returns_t of a price set prices as the relative change between the price at time $t+1$ and the previous price at time t , i.e.,

$$\text{returns}_t = \frac{\text{prices}_{t+1} - \text{prices}_t}{\text{prices}_t} \quad t = 1, \dots, N-1, \quad (7.2)$$

where N denotes the number of items in the list. The historical returns are used to estimate future returns. The time horizon used to compute returns is one day. After sorting the return list returns , we calculate the Value at Risk, VaR , as

$$\text{VaR} = \text{returns}' [\lfloor \alpha \cdot \text{length}(\text{returns}) \rfloor], \quad (7.3)$$

where $\text{returns}'$ is the ordered (sorted) list of returns and α denotes the confidence level. Thus, in a data set with 20 historical returns – with the first element located at position 1 and the last element located at position 20 – we select the return on position 19 (to be the

	Prices			Returns			Returns (sorted)	
	t	<i>hist</i>	<i>event</i>	t	<i>hist</i>	<i>event</i>	<i>hist</i>	<i>event</i>
	1	0.35	0.35	1	0.14	0.14	4.77	0.46
	2	0.40	0.40	2	0.07	0.07	0.54	0.14
	3	0.43	0.43	3	-0.05	-0.05	0.36	0.14
*	4	0.41	0.41	4	0.54	0.00	0.14	0.07
	5	0.63	0.41	5	-0.79	0.00	0.14	0.05
	6	0.13	0.41	6	4.77	0.00	0.07	0.03
	7	0.75	0.41	7	-0.44	0.00	0.05	0.00
	8	0.42	0.41	8	0.36	0.00	0.05	0.00
	9	0.57	0.41	9	0.05	0.46	0.03	0.00
	10	0.60	0.60	→ 10	-0.15	-0.15	→ -0.02	0.00
	11	0.51	0.51	11	-0.16	-0.16	-0.03	0.00
	12	0.43	0.43	12	-0.14	-0.14	-0.03	-0.02
	13	0.37	0.37	13	-0.03	-0.03	-0.05	-0.03
	14	0.36	0.36	14	0.14	0.14	-0.07	-0.03
	15	0.41	0.41	15	0.05	0.05	-0.08	-0.05
	16	0.43	0.43	16	-0.02	-0.02	-0.14	-0.07
	17	0.42	0.42	17	-0.07	-0.07	-0.15	-0.08
	18	0.39	0.39	18	0.03	0.03	-0.16	-0.14
	19	0.40	0.40	19	-0.08	-0.08	-0.44	-0.15
	20	0.37	0.37	20	-0.03	-0.03	-0.79	-0.16
	21	0.36	0.36					

Table 7.2: Example VaR calculation for returns with and without noise cleaning.

VaR) when using a confidence level of 0.95. Using Equation (7.3), we calculate VaR_{event} and VaR_{hist} using our adjusted method and the traditional method (i.e., the historical method without the improvements proposed by Boudoukh et al. (1998) and Hull and White (1998)), respectively. An example is given in Table 7.2. Here, the results of a VaR calculation are presented based on 21 prices – with and without cleaning – with an *event* occurring at $t = 4$ (denoted by a *) while using a *window* size of five. With a confidence interval of 95% (5% probability for the VaR definition in Section 7.2), this would result in a VaR of -0.44 or -0.15 (printed in bold font) for returns for the historical method or the event-based historical method, respectively (the VaR position in the returns is $95\% \cdot 20 = 19$). The observed difference stems from the proposed removal of noise inherently associated with events, i.e., the noise in prices generated from time $t = 4+1$ to time $t = 4+5$. These differences can then be evaluated by assessing the quality of both predicted values.

7.4 Evaluation

We employ various measures to evaluate the performance of the proposed historical VaR calculation for fixed-composition portfolios using a data set of stock rates and news events. First, we discuss our data, and then, we elaborate on the metrics used. Finally, we present our experimental results.

7.4.1 Data

In our experiments, we employ a data set stemming from the ViewerPro software, which, after the processing steps described in Section 7.3, contains news events and stock data collected on an hourly basis for 363 equities on weekdays during the year 2010. The data set consists of approximately 2,000 stock data points, 119 event types, and 50 - 75 event instances per equity. To evaluate the performance of the calculation, we predict the VaR_{event} and VaR_{hist} for 75% of our data set. The remaining 25% is used as a test set for comparing the predicted VaR with the actual VaR.

7.4.2 Metrics

In contrast to common approaches to evaluating VaR calculations, our approach does not employ the Kupiec test (Kupiec, 1995), as the test is statistically weak with small sample sizes (e.g., one year). As the data set we employed only covers 2010, we need different measures that provide insight into the effectiveness of our proposed event-based

approach. Therefore, to analyze the number of equities for which our adjusted event-based historical method provides better-quality predictions than the traditional historical method, we measure each method's squared error SE for equity c . The squared error is defined as follows:

$$SE_c = (VaR_{c,actual} - VaR_{c,predicted})^2 , \quad (7.4)$$

where $VaR_{c,actual}$ and $VaR_{c,predicted}$ represent equity c 's actual VaR measured in our test set and the predicted VaR based on our training set, respectively, and $VaR_{c,predicted}$ is one of VaR_{event} or VaR_{hist} .

Subsequently, we combine the squared errors into the mean squared error (MSE) for the historical method and the event-based historical method, i.e., MSE_{hist} and MSE_{event} . The MSE is defined as the summation of the squared errors (SE) of all equities $c \in C$ divided by the number of equities, i.e.,

$$MSE = \frac{\sum_{c \in C} SE_c}{|C|} , \quad (7.5)$$

where $|C|$ denotes the total number of equities in set E (363).

Additionally, we evaluate the number of times both methods outperform the other, i.e., OPT (OutPerformed Total). To do this, we compare the computed squared errors $SE_{c,hist}$ and $SE_{c,event}$ for all equities $c \in C$:

$$OPT_{hist,event} = \sum_{c \in C} O(SE_{c,hist}, SE_{c,event}) , \quad (7.6)$$

$$OPT_{event,hist} = \sum_{c \in C} O(SE_{c,event}, SE_{c,hist}) , \quad (7.7)$$

$$O(X, Y) = \begin{cases} 1 & \text{if } X < Y \\ 0 & \text{else} \end{cases} . \quad (7.8)$$

In our experiments, we compare the MSE and OPT for the traditional and event-based VaR calculation methods, both determined for the full event data set and for a data set containing only the rare events, using a time window of 8 hours (determined based on initial estimates). We then determine the optimal time window size by observing event-based VaR calculation plots of (normalized) MSE and OPT values for time windows ranging from 1 to 24 (i.e., three working days of 8 hours, which is the maximum effect of a news event (Kalev et al., 2004)). In addition, we consider the number of overconfident predictions ($CONF$) of all equities $c \in C$, which is calculated as follows:

$$CONF = \sum_{c \in C} Q(VaR_{c,predicted}, VaR_{c,actual}) , \quad (7.9)$$

$$Q(X, Y) = \begin{cases} 1 & \text{if } X > Y \\ 0 & \text{else} \end{cases} , \quad (7.10)$$

where $VaR_{e,predicted}$ represents the predicted VaR_{event} for equity c based on our adjusted data set (containing only the rare events). To optimize the size of the time window, for each window size, we normalize its associated MSE , $OPT_{event,hist}$, and $CONF$ using min-max normalization based on the previously computed values for the full range of window sizes, resulting in MSE' , OPT' , and $CONF'$, respectively. Subsequently, we subtract $CONF'$ from the computed difference between OPT' and MSE' , and normalize the result using min-max normalization to obtain a final score S that is to be maximized:

$$S = (OPT' - MSE') - CONF' . \quad (7.11)$$

Finally, the significance of the results is assessed by performing a two-sample one-tailed Student's t-test on the sets of individual squared errors SE_{hist} and SE_{event} for our optimal configuration. To do this, we use a significance level of 0.05 to reject the null hypothesis that there is no difference between the measured MSE values.

7.4.3 Results

Table 7.3 presents the MSE and OPT values for the traditional and event-based historical VaR calculation methods. When using all (i.e., regular and rare) events and stock rates on an hourly basis and when employing a time window of 8 hours, we observe a decrease of 21.44% in terms of MSE when accounting for event-generated noise in our stock data. The event-based VaR calculation method outperforms the traditional historical method in 76.18% of cases. Removing only the noise generated by rare events yields an additional improvement over the previous results. In 79.12% of the cases, event-based historical VaR calculation outperforms the historical method. The MSE is 26.37% lower for the event-based historical VaR calculations when only rare events are considered.

Measure	All events		Rare events	
	<i>hist</i>	<i>event</i>	<i>hist</i>	<i>event</i>
<i>MSE</i>	1.1234E-05	8.8254E-06	1.1234E-05	8.2717E-06
<i>OPT</i>	81	259	71	269

Table 7.3: Comparison of the performance of traditional and event-based historical VaR calculations using a window of 8 hours.

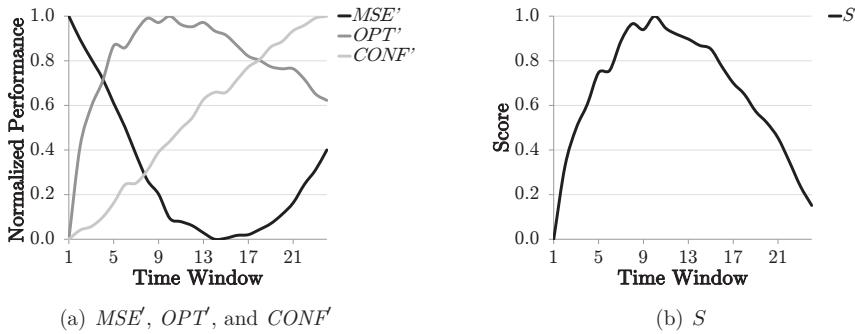


Figure 7.3: Performance of event-based VaR prediction models.

As illustrated in Figure 7.3, we obtain an optimized window size of 10. Although the MSE and OPT scores have a clear minimum and maximum, respectively, for our data set, the number of overconfident predictions becomes increasingly large when the time window is enlarged. Thus, when the differences between the OPT and MSE scores for various time window sizes are small, one should focus on minimizing the number of overconfident predictions, which results in small window sizes.

As shown in Table 7.4, utilizing a window of 10 instead of 8 hours on a data set with rare events indeed yields improvements when compared to the results shown in Table 7.3. The MSE of our event-based historical VaR prediction models decreases by 31.78% over the traditional historical VaR prediction method's MSE , and event-based VaR prediction outperforms the historical method in 77.71% of cases. Alternatively, even greater improvements are achieved when we determine the optimal cleaning window for each event type separately, with large differences in stock prices (we employ a threshold of 50.00% in our experiments) after an event occurrence indicating noise that should be cleaned and small differences indicating that the market has returned to normal. The measured MSE values decrease by 35.55%, and 75.59% of the event-based VaR predictions outperform the historical method.

Measure	window = 10		window = event-based	
	hist	event	hist	event
MSE	1.1234E-05	7.6633E-06	1.1234E-05	7.2402E-06
OPT	76	265	83	257

Table 7.4: Comparison of the performance of traditional and event-based historical VaR calculation with rare events.

To assess the significance of the measured MSE improvement of 35.55%, we perform a paired two-sample one-tailed t-test based on SE_{hist} and SE_{event} , containing squared errors for all equities. We obtain a p -value of 0.0027 and reject the null hypothesis that there is no difference between the measured MSE values at a significance level of 0.05. Thus, the proposed event-based historical VaR calculation method (using rare events and event-based window sizes) produces more reliable VaR predictions than the traditional method.

7.5 Conclusions

VaR is one of the most widely used risk assessment measures in the financial world, and has even become a requirement for regulatory purposes. The historical VaR calculation is a popular method for computing VaR in real time. The historical stock return data used in order to calculate VaR may, however, be sensitive to outliers caused by seldom-occurring news events that occur during the sampled period. Therefore, in this study, we have proposed a way to enhance the prediction of VaR based on historical data by removing disturbances induced by such events. Removing such disturbances would enable practitioners to make better predictions of risk in terms of distributions of expected future returns.

Based on a data set of stock rates and news events obtained using the proprietary ViewerPro software, we have identified news events that were likely to generate noise. This event-generated noise was subsequently removed from the stock rates. Based on our experiments, in which we evaluated various cleaning window sizes, we can conclude that VaR can be improved with news as an additional input. When using an arbitrary cleaning window of 8 hours, we observed that event-based historical VaR calculations produced more accurate results than traditional historical VaR calculations, resulting in lower MSE scores. When only rare events (identified using a Poisson distribution) were considered, the decrease in MSE increased from 21.44% to 26.37%, and our new method outperformed the traditional method more often (79.12% versus 76.18% of the cases). Moreover, we have optimized the cleaning window to 10 hours (when considering only rare events), resulting in an MSE improvement of 31.78% and event-based VaR calculation outperforming the traditional method in 77.71% of cases. Alternatively, we considered a per-event cleaning window optimization strategy that demonstrated a significant MSE improvement of 35.55% compared to the traditional historical method, outperforming the latter in 75.59% of cases.

In future work, we propose to investigate how to account for the type of news event in our VaR method, which could affect the influence of an event on equity prices (e.g., mergers could generate more noise than quarterly profit announcements). Another future research direction is related to accounting for general stock market events, such as financial crises, instead of company-specific news only. We would also like to build a real-life market simulation for our improved historical VaR method.

Chapter 8

Conclusions and Outlook

In this dissertation, we have investigated techniques for event extraction from news, and devised methods to incorporate events into common financial applications. In our search for an answer to the central problem statement – i.e., how to semi-automatically and accurately identify financial events in news messages, and how to effectively use such extracted events in financial applications – we have touched upon various aspects of event extraction, such as text processing, expert knowledge incorporation, and knowledge base updating, as well as two financial applications, i.e., automated trading and risk analysis. Our most important findings are discussed next. Moreover, we put the disseminated research into perspective and provide additional directions for future research.

8.1 Concluding Remarks

We have reviewed the current body of literature on event extraction, where we distinguished between data-driven, knowledge-driven, and hybrid approaches to event extraction. An analysis on a set of qualitative dimensions – i.e., the amount of required data, knowledge, and expertise, as well as the interpretability of the results and the required development and execution times – showed the clear distinction between the three approaches, yet underlined that the choice for a suitable extraction technique for a specific application is anything but arbitrary. Such a decision depends on one's needs on the one hand, and the amount of available data, knowledge, and expertise on the other hand. Additionally, we identified two major application areas of events, namely biomedics (for identifying molecular events, protein bindings, etc.) and news digestion (e.g., for summarization, recommendation, border security, or even automated trading), suggesting the wide applicability of events in business. Moreover, we determined the most important open research issues, which primarily relate to (the lacking awareness of) limitations of

specific extraction techniques, domain-dependencies affecting flexibility and scalability, and the complicating factors of human ambiguity and diversity. Last, we identified the main extraction tools for various programming languages like Java and Python, such as GATE, NLTK, and LingPipe, and we researched the common practices and peculiarities of event extraction evaluation, which we took into consideration throughout the rest of the dissertation.

Based on the insights gained from our extensive literature survey, we created a primarily knowledge-driven event extraction pipeline, which extracts financial events from news articles, and annotates these with meta-data at a speed that enables real-time use. The framework builds partially on existing components, and we have additionally developed our own, high-performance, knowledge-driven components. Through their interaction with a domain-specific ontology, our novel, semantically-enabled components constitute a feedback loop which fosters future reuse of acquired knowledge in the event detection process. This renders the pipeline to be competitive when compared to its state-of-the-art alternatives, as we have witnessed high precision, recall, and F_1 scores that were obtained within subsecond processing times.

We further investigated the incorporation of domain knowledge into event extraction systems through the development of a knowledge-based, lexico-semantic pattern language for defining extraction rules. The language, which makes use of lexical, syntactic, and semantic elements, greatly benefits from a well-defined domain ontology by inferencing on its specified concepts and relations. The pattern language is comparable to other, existing information extraction languages, yet has a competitive advantage because of its unique combination of high expressivity and ease of use due to the employment of semantic elements, as well as its applicability to more complex (event) extraction tasks. Our evaluation based on development times, precision, recall, and F_1 scores on a financial and political data set demonstrated the superiority of our language over its existing counterparts. Moreover, we have researched an evolutionary approach to rule learning that is based on genetic programming. Initial experiments have shown that such a learning algorithm outperforms the process of manual rule construction, and hence could potentially boost the acceptance of such knowledge-driven patterns by a broader audience, because of the decreased costs of involving domain experts.

Subsequently, we have examined the automation of the time and labor-intensive process of knowledge base updating. For this, we presented and implemented an event-driven, trigger-based extension to the existing update language OUL, which is aimed specifically at ontologies that form the core of semantically-enabled, knowledge-driven event extraction frameworks like the one presented in this dissertation. Moreover, we focused on the

support for various execution models, enabling for instance deferred, looped, or chained execution of updates which are automatically triggered after an event occurrence. This fosters the improvement of overall knowledge base quality, by ensuring the ontology is up-to-date with the latest discovered (or deduced) facts, which can immediately be used in the subsequent (news) text processing iterations.

Last, we researched two applications in the finance domain in which we introduced events as additional signals for their core computations. In our first application, we looked into the construction of trading rules, which are traditionally based on numerical technical trading indicators (e.g., stock price averages). After converting events extracted from news into trading signals, we devised a genetic programming approach in order to learn trading rules, while optimizing their revenue across multiple time horizons. Our analysis revealed that for both considered financial data sets and for different time horizons, optimal trading strategies included an event-based signal, which is a clear indication of the added value of extracted events for predictive purposes. Therefore, we additionally looked into the prediction of Value at Risk (VaR), in which we considered rare events to be a possible cause of price trend disruptions due to the acknowledged sensitivity of stock prices to emerging news. Our data set was cleaned from event-generated disturbances, and the quality of (historical) VaR calculations was compared against the performance of the calculations on the original data set. Our experiments demonstrated a substantial improvement when removing the event-generated disturbances from the stock prices for a small, optimized time window.

The latter findings all contribute to the conclusion that we are able to accurately identify financial events through a semi-automatic, knowledge-driven event extraction system consisting of semantically-enabled components and exploiting lexico-semantic patterns. Such extracted events can subsequently be used as additional, high-quality, inputs in various financial applications, such as algorithmic trading and VaR computations, where in both cases we have seen that results demonstrably improved when taking into consideration the discovered news events.

8.2 Outlook

Although currently, we relied on manually constructed and semi-automatically maintained ontologies, we envision future event extraction frameworks to be based on automatically constructed taxonomies (de Knijff et al., 2011, 2013; Meijer et al., 2014; Vandic et al., 2014), or even on centrally maintained, large, widely accessible, and linked open databases such as DBpedia, Wikidata, and Freebase. The research in this area has matured to a

level at which it is directly applicable to event extraction frameworks in order to further stimulate a fully automated operation, in which expert knowledge is a driving, yet a less costly, factor.

The field of event extraction is gradually moving toward the inclusion of time (Frasicar et al., 2010; Milea et al., 2008, 2012a,b) and space (Ho et al., 2012; Hogenboom et al., 2010a,c) dimensions, yet in our endeavors they are not yet taken into consideration due to their added complexity. Consequently, we do not explicitly track events (stories) over time (Verheij et al., 2012a), and hence we do not detect sequencing of events, but we assume events to be independent of one another. However, developments in storylines can still be modeled by capturing the various stages in fine-grained events, for instance by distinguishing between announcements and realisations of specific events. Future applications, however, will have a principal, more robust, and thorough way of dealing with temporal and spatial aspects, eliminating the need for overly detailed event definitions to capture the spatio-temporal dimensions, and hence fostering a more intuitive approach for complex event extraction. Moreover, we envision a full incorporation of spatio-temporal aspects, not only by connecting these aspects to events, but also by exploiting this information in reasoning and discovering new events. In other words, additional information on events will not only be collected, but will also be used effectively both in subsequent tasks and in future collection phases.

Another ongoing development in the studied field is the increased attention for sentiment (Bal et al., 2011; Feldman, 2013), or even the distinction between rumors and facts. Even though we do not consider the latter novelties, we initially account for these issues by delivering a semi-automatic approach to event extraction, in which experts approve and/or assign impacts to event outputs. We do, however, envision an increase in the amount of available complex components for sentiment analysis and other advanced text processing tasks, which can be connected to our already existing (semantically-enabled) event extraction components in future work. For instance, linking sentiment to discovered events, can provide for an automatic way to discover the influence of an event type on stock prices.

Most applications of event extraction, such as the ones presented in this dissertation (i.e., automated trading and VaR assessments), are fully automated and hence are highly useful in automated environments such as high-frequency trading. The proposed techniques primarily focused on improving the inputs of algorithms, and not so much on improving the algorithms themselves. For future real-world applications, it is of primary importance to optimize efficiency of the driving semi-automatic event extraction processes, or even to fully automate the latter procedures, as these currently form the bottleneck

of event-driven applications. In the near future, the research emphasis will shift to more efficient or intelligent event usages in existing algorithms, in order to optimize the actual computations rather than the event acquisition procedures. For instance, trading rules can be contextualized, as events can have a different associated importance in the rules, based on the evolution of stock prices.

Given the considerations above, we envision a bright future for event extraction. Its advantageous applications, especially in financial computations, can be of good use in automated and semi-automated environments. Also, due to the developments in the closely related fields of text mining and information extraction, as well as the recent advances in Semantic Web technologies, future event-driven frameworks will turn out to be powerful, intelligent, and profitable tools that can be employed in a wide variety of applications.

Bibliography

- S. B. Achelis. *Technical Analysis from A to Z*. McGraw-Hill, 2nd edition, 2000.
- J. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn. Open User Profiles for Adaptive News Systems: Help or Harm? In C. Williamson, M. E. Zurko, P. Patel-Schneider, and P. Shenoy, editors, *16th International Conference on World Wide Web (WWW 2007)*, pages 11–20. ACM, 2007.
- F. Allen and R. Karjalainen. Using Genetic Algorithms to Find Technical Trading Rules. *Journal of Financial Economics*, 51(2):245–271, 1999.
- T. G. Andersen, T. Bollerslev, F. X. Diebold, and H. Ebens. The Distribution of Stock Return Volatility. *Journal of Financial Economics*, 61(1):43–76, 2001.
- P. J. Angeline. Subtree Crossover: Building Block Engine or Macromutation? In J. R. Koza, K. Deb, M. Dorigo, David B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, editors, *2nd Annual Conference on Genetic Programming (GP 1997)*, pages 9–17. Morgan Kaufmann, 1997.
- W. Antweiler and M. Z. Frank. Do US Stock Markets Typically Overreact to Corporate News Stories? From: <http://dx.doi.org/10.2139/ssrn.878091>, 2006.
- C. Aone and M. Ramos-Santacruz. REES: A Large-Scale Relation and Event Extraction System. In *6th Applied Natural Language Processing Conference (ANLP 2000)*, pages 76–83. Association for Computational Linguistics, 2000.
- L. Ardissono, L. Console, and I. Torre. An Adaptive System for the Personalized Access to News. *AI Communications*, 14(3):129–147, 2001.
- E. Arendarenko and T. Kakkonen. Ontology-Based Information and Event Extraction for Business Intelligence. In A. Ramsay and G. Agre, editors, *15th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA 2012)*,

- volume 7557 of *Lecture Notes in Computer Science*, pages 89–102. Springer Berlin Heidelberg, 2012.
- P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent Measures of Risk. *Mathematical Finance*, 9(3):203–228, 1999.
- F. Asche, R. E. Dahl, and A. Oglend. Value-at-Risk: Risk Assessment for the Portfolio of Oil and Gas Producers. UiS Working Papers in Economics and Finance 2013/3, University of Stavanger, 2013.
- M. Atkinson, J. Piskorski, H. Tanev, E. van der Goot, R. Yangarber, and V. Zavarella. Automated Event Extraction in the Domain of Border Security. In P. Daras and O. M. Ibarra, editors, *User Centric Media*, volume 40 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 321–326. Springer Berlin Heidelberg, 2009.
- M. Atkinson, M. Du, J. Piskorski, H. Tanev, R. Yangarber, and V. Zavarella. Techniques for Multilingual Security-Related Event Extraction from Online News. In A. Przeździecki, M. Piasecki, K. Jassem, and P. Fuglewicz, editors, *Computational Linguistics*, volume 458 of *Studies in Computational Intelligence*, chapter 9, pages 163–186. Springer Berlin Heidelberg, 2013.
- R. Baeza-Yates. Big Data or Right Data? In L. Bravo and M. Lenzerini, editors, *7th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW 2013)*, volume 1087 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.
- D. Bal, M. Bal, A. van Bunningen, A. Hogenboom, **F. Hogenboom**, and F. Frasincar. Sentiment Analysis with a Multilingual Pipeline. In A. Bouguettaya, M. Hauswirth, and L. Liu, editors, *12th International Conference on Web Information System Engineering (WISE 2011)*, volume 6997 of *Lecture Notes in Computer Science*, pages 129–142. Springer, 2011.
- S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language Reference – W3C Recommendation 10 February 2004, 2004. From: <http://www.w3.org/TR/owl-ref/>.
- A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71, 1996.
- L. Berke. Algorithms Find Their Rhythm with Broad, Growing Base. *Traders Magazine Supplement*, pages 74–75, 2007.

- T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.
- C. Best, J. Piskorski, B. Pouliquen, R. Steinberger, and H. Tanev. Automating Event Extraction for the Security Domain. In H. Chen and C. C. Yang, editors, *Intelligence and Security Informatics*, volume 135 of *Studies in Computational Intelligence*, chapter 2, pages 17–43. Springer Berlin Heidelberg, 2008.
- D. Billsus and M. J. Pazzani. A Personal News Agent that Talks, Learns and Explains. In *3rd International Conference on Autonomous Agents (Agents 1999)*, pages 268–275. ACM, 1999.
- J. Björne, F. Ginter, S. Pyysalo, J. Tsujii, and T. Salakoski. Complex Event Extraction at PubMed Scale. *Bioinformatics*, 26(12):i382–i390, 2010.
- W. J. Black, J. McNaught, A. Vasilakopoulos, K. Zervanou, B. Theodoulidis, and F. Rinaldi. CAFETIERE: Conceptual Annotations for Facts, Events, Terms, Individual Entities, and RElations. Technical Report TR-U4.3.1, Department of Computation, UMIST, Manchester, 2005.
- S. Boag, D. Chamberlin, M. F. Fernandez, D. Florescu, J. Robie, and J. Simeon. XQuery 1.0: An XML Query Language (Second Edition) – W3C Recommendation 14 December 2010, 2010. From: <http://www.w3.org/TR/xquery/>.
- K.-A. Bonnier and R. F. Bruner. An Analysis of Stock Price Reaction to Management Change in Distressed Firms. *Journal of Accounting and Economics*, 11(1):95–106, 1989.
- C. Borg, M. Rosner, and G. J. Pace. Automatic Grammar Rule Extraction and Ranking for Definitions. In *7th International Conference of Language Resources and Evaluation (LREC 2010)*. European Language Resources Association, 2010.
- G. Bormetti, M. E. De Giuli, D. Delpini, and C. Tarantola. Bayesian Value-at-Risk with Product Partition Models. *Quantitative Finance*, 12(5):769–780, 2012.
- J. Borsje, L. Levering, and F. Frasincar. Hermes: A Semantic Web-Based News Decision Support System. In *23rd Annual ACM Symposium on Applied Computing*, pages 2415–2420. ACM, 2008.
- J. Borsje, **F. Hogenboom**, and F. Frasincar. Semi-Automatic Financial Events Discovery Based on Lexico-Semantic Patterns. *International Journal of Web Engineering and Technology*, 6(2):115–140, 2010.

- J. Boudoukh, M. Richardson, and R. F. Whitelaw. The Best of Both Worlds: A Hybrid Approach to Calculating Value at Risk. *Risk*, 11(5):64–67, 1998.
- T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible Markup Language (XML) – W3C Recommendation 26 November 2008, 2008. From: <http://www.w3.org/TR/2008/REC-xml-20081126/>.
- D. Brickley and R. Guha. RDF Vocabulary Description Language 1.0: RDF Schema – W3C Recommendation 10 February 2004, 2004. From: <http://www.w3.org/TR/rdf-schema/>.
- E. Brill. A Simple Rule-Based Part of Speech Tagger. In *3rd Conference on Applied Natural Language Processing (ANLP 1992)*, pages 152–155. Association for Computational Linguistics, 1992.
- W. A. Brock, J. Lakonishok, and B. LeBaron. Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. *Journal of Finance*, 47(5):1731–1764, 1992.
- H. Byström. News Aggregators, Volatility and the Stock Market. *Economics Bulletin*, 29(4):2673–2682, 2009.
- M. Capelle, M. Moerland, F. Frasincar, and **F. Hogenboom**. Semantics-Based News Recommendation. In R. Akerkar, C. Bădică, and D. Dan Burdescu, editors, *2nd International Conference on Web Intelligence, Mining and Semantics (WIMS 2012)*. ACM, 2012.
- M. Capelle, **F. Hogenboom**, A. Hogenboom, and F. Frasincar. Semantic News Recommendation Using WordNet and Bing Similarities. In S. Y. Shin and J. C. Maldonado, editors, *28th Symposium on Applied Computing (SAC 2013), The Semantic Web and its Application Track*, pages 296–302. ACM, 2013.
- P. Capet, T. Delavallade, T. Nakamura, A. Sandor, C. Tarsitano, and S. Voyatzis. A Risk Assessment System with Automatic Extraction of Event Types. In Z. Shi, E. Mercier-Laurent, and D. Leake, editors, *Intelligent Information Processing IV*, volume 288 of *IFIP Advances in Information and Communication Technology*, chapter 27, pages 220–229. Springer Boston, 2008.
- A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr., and T. M. Mitchell. Coupled Semi-Supervised Learning for Information Extraction. In *3rd International Conference on Web Search and Data Mining (WSDM 2010)*, pages 101–110. ACM, 2010.

- M. Castellanos, C. Gupta, S. Wang, and U. Dayal. Leveraging Web Streams for Contractual Situational Awareness in Operational BI. In F. Daniel, L. M. L. Delcambre, F. Fotouhi, I. Garrigós, G. Guerrini, J.-N. Mazón, M. Mesiti, S. Müller-Feuerstein, J. Trujillo, T. M. Truta, B. Volz, E. Waller, L. Xiong, and E. Zimányi, editors, *International Workshop on Business intelligenceE and the WEB (BEWEB 2010) in conjunction with EDBT/ICDT 2010 Joint Conference*, pages 1–8. ACM, 2010.
- M. Cecchini, H. Aytug, G. J. Koehler, and P. Pathak. Making Words Work: Using Financial Text as a Predictor of Financial Events. *Decision Support Systems*, 50(1): 64–175, 2010.
- D. D. Chamberlin and R. F. Boyce. SEQUEL: A Structured English Query Language. In R. Rustin, editor, *1974 ACM SIGMOD Workshop on Data Description, Access and Control*, volume 1, pages 249–264. ACM, 1974.
- W. S. Chan. Stock Price Reaction to News and No-News: Drift and Reversal After Headlines. *Journal of Financial Economics*, 70(2):223–260, 2003.
- C.-H. Chang, M. Kayed, M. R. Girgis, and K. Shaalan. A Survey of Web Information Extraction Systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10): 1411–1428, 2006.
- M. Chen, C. Zhang, and S.-C. Chen. Semantic Event Extraction Using Neural Network Ensembles. In *1st IEEE International Conference on Semantic Computing (ICSC 2007)*, pages 575–580. IEEE Computer Society, 2007.
- H.-W. Chun, Y.-S. Hwang, and H.-C. Rim. Unsupervised Event Extraction from Biomedical Literature Using Co-occurrence Information and Basic Patterns. In K.-Y. Su, J. Tsujii, J.-H. Lee, and O. Y. Kwong, editors, *1st International Joint Conference on Natural Language Processing (IJCNLP 2004)*, volume 3248 of *Lecture Notes in Computer Science*, pages 777–786. Springer Berlin Heidelberg, 2004.
- C. Clark. How to Keep Markets Safe in the Era of High-Speed Trading. In *Essays on Issues*, number 303 in Chicago Fed Letter. The Federal Reserve Bank of Chicago, 2012.
- J. Clark and S. DeRose. XML Path Language (XPath) – W3C Recommendation 16 November 1999, 1999. From: <http://www.w3.org/TR/xpath/>.
- K. B. Cohen, K. Verspoor, H. L. Johnson, C. Roeder, P. V. Ogren, W. A. Baumgartner, Jr., E. White, H. Tipney, and L. Hunter. High-Precision Biological Event Extraction

- with a Concept Recognizer. In *Workshop on BioNLP: Shared Task at 47th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2009)*, pages 50–58. Association for Computational Linguistics, 2009.
- J. Cowie and W. Lehnert. Information Extraction. *Communications of the ACM*, 39(1):80–91, 1996.
- H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36(2):223–254, 2002.
- H. Cunningham, D. Maynard, and V. Tablan. JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS-00-10, Department of Computer Science, University of Sheffield, November 2000.
- H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 168–175. Association for Computational Linguistics, 2002.
- A. Cybulski and P. Vossen. Historical Event Extraction from Text. In K. Zervanou and P. Lendvai, editors, *5th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2011) at 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pages 39–43. Association for Computational Linguistics, 2011.
- S. R. Das and M. Y. Chen. Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. *Management Science*, 53(9):1375–1388, 2007.
- J. de Knijff, K. Meijer, **F. Hogenboom**, and F. Frasincar. Word Sense Disambiguation for Automatic Taxonomy Construction from Text-Based Web Corpora. In A. Bouguettaya, M. Hauswirth, and L. Liu, editors, *12th International Conference on Web Information System Engineering (WISE 2011)*, volume 6997 of *Lecture Notes in Computer Science*, pages 241–248. Springer, 2011.
- J. de Knijff, F. Frasincar, and **F. Hogenboom**. Domain Taxonomy Learning from Text: The Subsumption Method versus Hierarchical Clustering. *Data & Knowledge Engineering*, 83(1):54–69, 2013.
- B. Decadt, V. Hoste, W. Daelemans, and A. van den Bosch. GAMBL, Genetic Algorithm Optimization of Memory-Based WSD. In R. Mihalcea and P. Edmonds, editors,

- 3rd ACL/SIGLEX International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (senseval-3), pages 108–112. Association for Computational Linguistics, 2004.
- E. W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959.
- D. N. Dimitrakopoulos, M. G. Kavussanos, and S. I. Spyrou. Value at Risk Models for Volatile Emerging Markets Equity Portfolios. *The Quarterly Review of Economics and Finance*, 50(4):515–526, 2010.
- J. Domingue and E. Motta. PlanetOnto: From News Publishing to Integrated Knowledge Management Support. *IEEE Intelligent Systems*, 15(3):26–32, 2000.
- B. Drury and J. J. Almeida. Identification of Fine Grained Feature Based Event and Sentiment Phrases from Business News Stories. In R. Akerkar, editor, *1st International Conference on Web Intelligence, Mining and Semantics (WIMS 2011)*. ACM, 2011.
- J. E. Engelberg and C. A. Parsons. The Causal Impact of Media in Financial Markets. *Journal of Finance*, 66(1):67–97, 2009.
- O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised Named-Entity Extraction From The Web: An Experimental Study. *Artificial Intelligence*, 165(1):91–134, 2005.
- O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open Information Extraction from the Web. *Communications of the ACM*, 51(12):68–74, 2008.
- S. G. Ewalds, M. B. J. Schauten, and O. W. Steenbeek. De Informatiewaarde van Kwartaalcijfers. *Maandblad voor Accountancy en Bedrijfseconomie*, 1(7/8):333–341, 2000.
- E. F. Fama. The Behavior of Stock-Market Prices. *Journal of Business*, 38(1):34–105, 1965.
- W. Fan, P. Pathak, and L. Wallace. Nonlinear Ranking Function Representations in Genetic Programming-Based Ranking Discovery for Personalized Search. *Decision Support Systems*, 42(3):1338–1349, 2006.
- R. Feldman. Techniques and Applications for Sentiment Analysis. *Communications of the ACM*, 56(4):82–89, 2013.

- C. Fellbaum. WordNet: An Electronic Lexical Database. *Computational Linguistics*, 25(2):292–296, 1998.
- F. Frasincar, J. Borsje, and L. Levering. A Semantic Web-Based Approach for Building Personalized News Services. *International Journal of E-Business Research*, 5(3):35–53, 2009.
- F. Frasincar, V. Milea, and U. Kaymak. tOWL: Integrating Time in OWL. In R. De Virgilio, F. Giunchiglia, and L. Tanca, editors, *Semantic Web Information Management: A Model-Based Perspective*, chapter 11, pages 225–246. Springer, 2010.
- F. Frasincar, J. Borsje, and **F. Hogenboom**. Personalizing News Services Using Semantic Web Technologies. In I. Lee, editor, *E-Business Applications for Product Development and Competitive Growth: Emerging Technologies*, chapter 13, pages 261–289. IGI Global, 2011a.
- F. Frasincar, W. IJntema, F. Goossen, and **F. Hogenboom**. A Semantic Approach for News Recommendation. In M. E. Zorrilla, J.-N. Mazón, Óscar Ferrández, I. Garrigós, F. Daniel, and J. Trujillo, editors, *Business Intelligence Applications and the Web: Models, Systems and Technologies*, chapter 5, pages 102–121. IGI Global, 2011b.
- L. T. F. Gamut. *Introduction to Logic*, volume 1 of *Logic, Language, and Meaning*. The University of Chicago Press, 1991.
- J. Gibbons. Nonparametric Statistical Inference. *Technometrics*, 28(3):275, 1986.
- R. Goonatilake and S. Herath. The Volatility of the Stock Market and News. *International Research Journal of Finance and Economics*, 3(11):53–65, 2007.
- F. Goossen, W. IJntema, F. Frasincar, **F. Hogenboom**, and U. Kaymak. News Personalization using the CF-IDF Semantic Recommender. In R. Akerkar, editor, *1st International Conference on Web Intelligence, Mining and Semantics (WIMS 2011)*. ACM, 2011.
- R. Grishman and B. Sundheim. Message Understanding Conference – 6: A Brief History. In *16th International Conference on Computational Linguistics (COLING 1996)*, volume 1, pages 466–471. Association for Computational Linguistics, 1996.
- B. M. Gross. *The Managing of Organizations: The Administrative Struggle*, volume 1. Free Press of Glencoe, 1964.

- T. R. Gruber. A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.
- N. Guarino and C. A. Welty. Evaluating Ontological Decisions with OntoClean. *Communications of the ACM*, 45(2):61–65, 2002.
- M. A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *14th Conference on Computational Linguistics (COLING 1992)*, volume 2, pages 539–545, 1992.
- M. A. Hearst. Automated Discovery of WordNet Relations. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database and Some of its Applications*, chapter 5, pages 131–151. MIT Press, 1998.
- T. Hellstrom and K. Holmstrom. Parameter Tuning in Trading Algorithms using ASTA. In *6th International Conference Computational Finance (CF 1999)*, pages 343–357. MIT Press, 1999.
- S.-S. Ho, M. Lieberman, P. Wang, and H. Sarnet. Mining Future Spatiotemporal Events and their Sentiment from Online News Articles for Location-Aware Recommendation System. In C.-Y. Chow and S. Shekhar, editors, *1st ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems (MobiGIS 2012)*, pages 25–32. ACM, 2012.
- A. Hogenboom, **F. Hogenboom**, F. Frasincar, U. Kaymak, O. van der Meer, and K. Schouten. Detecting Economic Events Using a Semantics-Based Pipeline. In A. Hameurlain, S. W. Liddle, K.-D. Schewe, and X. Zhou, editors, *22nd International Conference on Database and Expert Systems Applications (DEXA 2011)*, volume 6860 of *Lecture Notes in Computer Science*, pages 440–447. Springer, 2011a.
- A. Hogenboom, E. Nieuwenhuijse, **F. Hogenboom**, and F. Frasincar. RCQ-ACS: RDF Chain Query Optimization Using an Ant Colony System. In N. Zhong, Z. Gong, Y. ming Cheung, P. Lingras, P. S. Szczepaniak, and E. Suzuki, editors, *The 2012 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2012)*, pages 74–81. IEEE Computer Society, 2012a.
- A. Hogenboom, F. Frasincar, and U. Kaymak. Ant Colony Optimization for RDF Chain Queries for Decision Support. *Expert Systems with Applications*, 40(5):1555–1563, 2013a.

- A. Hogenboom, **F. Hogenboom**, F. Frasincar, K. Schouten, and O. van der Meer. Semantics-Based Information Extraction for Detecting Economic Events. *Multimedia Tools and Applications*, 64(1):27–52, 2013b.
- F. Hogenboom.** Financial Events Recognition in Web News for Algorithmic Trading. In S. Castano, P. Vassiliadis, L. V. Lakshmanan, and M. L. Lee, editors, *9th International Workshop on Web Information Systems Modeling (WISM 2012) at 31st International Conference on Conceptual Modeling (ER 2012)*, volume 7518 of *Lecture Notes in Computer Science*, pages 368–377. Springer, 2012.
- F. Hogenboom**, F. Frasincar, and U. Kaymak. A Survey of Approaches on Mining the Structure from Unstructured Data. In *Dutch-Belgian Database Day 2009 (DBDBD 2009)*, 2009. From: <http://www.wis.ewi.tudelft.nl/index.php/dbdbd2009>.
- F. Hogenboom**, B. Borgman, F. Frasincar, and U. Kaymak. Spatial Knowledge Representation on the Semantic Web. In *4th IEEE International Conference on Semantic Computing (ICSC 2010)*, pages 252–259. IEEE Computer Society, 2010a.
- F. Hogenboom**, F. Frasincar, and U. Kaymak. An Overview of Approaches to Extract Information from Natural Language Corpora. In M. van der Heijden, M. Hinne, W. Kraaij, M. van Kuppeveld, S. Verberne, and T. van der Weide, editors, *10th Dutch-Belgian Information Retrieval Workshop (DIR 2010)*, pages 69–70, 2010b. From: http://www.ru.nl/publish/pages/544689/proceedings_dir2010.pdf.
- F. Hogenboom**, F. Frasincar, and U. Kaymak. A Review of Approaches for Representing RCC8 in OWL. In W. Chu, W. E. Wong, M. J. Palakal, and C. Hung, editors, *25th Symposium On Applied Computing (SAC 2010), The Semantic Web and its Application Track*, pages 1444–1445. ACM, 2010c.
- F. Hogenboom**, A. Hogenboom, F. Frasincar, U. Kaymak, O. van der Meer, K. Schouten, and D. Vandic. SPEED: A Semantics-Based Pipeline for Economic Event Detection. In J. Parsons, M. Saeki, P. Shoval, C. C. Woo, and Y. Wand, editors, *29th International Conference on Conceptual Modeling (ER 2010)*, volume 6412 of *Lecture Notes in Computer Science*, pages 452–457. Springer Berlin Heidelberg, 2010d.
- F. Hogenboom**, V. Milea, F. Frasincar, and U. Kaymak. Graphically Querying RDF Using RDF-GL. In *Dutch-Belgian Database Day 2010 (DBDBD 2010)*, 2010e. From: <http://www.uhasselt.be/Documents/UHasselt/initiatieven/DBDBD-2010/HogenboomF.pdf>.

- F. Hogenboom**, V. Milea, F. Frasincar, and U. Kaymak. RDF-GL: A SPARQL-Based Graphical Query Language for RDF. In R. Chbeir, Y. Badr, A. Abraham, and A.-E. Hassanien, editors, *Emergent Web Intelligence: Advanced Information Retrieval, Advanced Information and Knowledge Processing*, chapter 4, pages 87–116. Springer London, 2010f.
- F. Hogenboom**, F. Frasincar, U. Kaymak, and F. de Jong. An Overview of Event Extraction from Text. In M. van Erp, W. R. van Hage, L. Hollink, A. Jameson, and R. Troncy, editors, *Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at 10th International Semantic Web Conference (ISWC 2011)*, volume 779 of *CEUR Workshop Proceedings*, pages 48–57. CEUR-WS.org, 2011b.
- F. Hogenboom**, F. Frasincar, U. Kaymak, and F. de Jong. News Recommendations using CF-IDF. In P. de Causmaecker, J. Maervoet, T. Messelis, K. Verbeeck, and T. Vermeulen, editors, *23rd Benelux Conference on Artificial Intelligence (BNAIC 2011)*, pages 397–398. Neveland, 2011c.
- F. Hogenboom**, M. de Winter, F. Frasincar, and A. Hogenboom. A News-Based Approach for Computing Historical Value-at-Risk. In J. Casillas, F. J. Martínez-López, and J. M. Corchado, editors, *1st International Symposium on Management Intelligent Systems (IS-MiS 2012)*, volume 171 of *Advances in Intelligent Systems and Computing*, pages 283–292. Springer, 2012b.
- F. Hogenboom**, M. de Winter, M. Jansen, A. Hogenboom, F. Frasincar, and U. Kaymak. Event-Based Historical Value-at-Risk. In *IEEE Computational Intelligence for Financial Engineering & Economics 2012 (CIFEr 2012)*, pages 66–72. IEEE Computer Society, 2012c.
- F. Hogenboom**, A. Hogenboom, and F. Frasincar. Semantics-Based Financial Event Detection. In T. Demeester, J. Deleu, L. Mertens, D. Plaetinck, A. de Moor, T. Hoang, T. Wauters, C. Develder, B. Vermeulen, and P. Demeester, editors, *12th Dutch-Belgian Information Retrieval Workshop (DIR 2012)*, pages 71–72, 2012d.
- F. Hogenboom**, W. IJntema, and F. Frasincar. Text-Based Information Extraction Using Lexico-Semantic Patterns. In J. W. H. M. Uiterwijk, N. Roos, and M. H. M. Winands, editors, *24th Benelux Conference on Artificial Intelligence (BNAIC 2012)*, pages 293–294. Océ Business Services, 2012e.

- F. Hogenboom**, J. Sangers, and F. Frasincar. Ontology Updating Driven by Events. In *Dutch-Belgian Database Day 2012 (DBDBD 2012)*, 2012f. From: <http://dbdbd.be/wp-content/uploads/9-Ontology-Updating-Driven-by-Events.pdf>.
- F. Hogenboom**, M. de Winter, F. Frasincar, and U. Kaymak. A News Event-Driven Approach for the Historical Value at Risk Method. *Expert Systems With Applications*, 2013c. To Appear.
- F. Hogenboom**, W. IJntema, and F. Frasincar. Learning Semantic Information Extraction Rules from News. In *Dutch-Belgian Database Day 2013 (DBDBD 2013)*, 2013d. From: <http://dbdbd.nl/wp-content/uploads/abstract8.pdf>.
- F. Hogenboom**, M. Capelle, and M. Moerland. News Recommendation using Semantics with the Bing-SF-IDF Approach. In J. Parsons and D. Chiu, editors, *10th International Workshop on Web Information Systems Modeling (WISM 2013) at 32nd International Conference on Conceptual Modeling (ER 2013)*, volume 8697 of *Lecture Notes in Computer Science*, pages 164–173. Springer, 2014a.
- F. Hogenboom**, M. Capelle, M. Moerland, and F. Frasincar. Bing-SF-IDF+: Semantics-Driven News Recommendation. In C.-W. Chung, A. Z. Broder, K. Shim, and T. Suel, editors, *23rd International World Wide Web Conference (WWW Companion 2014)*, pages 291–292. ACM, 2014b.
- F. Hogenboom**, F. Frasincar, U. Kaymak, and F. de Jong. A Survey of Event Extraction Methods from Text for Decision Support Systems. *Decision Support Systems*, 2014c. Under Review.
- F. Hogenboom**, D. Vandic, F. Frasincar, A. Verheij, and A. Kleijn. A Query Language and Ranking Algorithm for News Items in the Hermes News Processing Framework. *Science of Computer Programming*, 94, Part 1:32–52, 2014d.
- J. H. Holland. Genetic Algorithms. *Scientific American*, 267(1):66–72, 1992.
- M. Hollander and D. Wolfe. Nonparametric Statistical Methods. *Journal of the American Statistical Association*, 95(449):333, 2000.
- G. A. Holton. *Value at Risk: Theory and Practice*. Academic Press, 1st edition, 2003.
- P.-Y. Hsueh, P. Melville, and V. Sindhwani. Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria. In *Workshop on Active Learning for Natural Language Processing (ALNLP 2009) at 6th North American Chapter of the Association for Computational Linguistics (NAACL 2009)*, 2009.

- Computational Linguistics (HLT-NAACL 2009)*, pages 27–35. Association for Computational Linguistics, 2009.
- H. Huang and T.-H. Lee. Forecasting Value-at-Risk Using High-Frequency Information. *Econometrics*, 1(1):127–140, 2013.
- J. C. Hull. *Options, Futures, and Other Derivatives*. Pearson Education Limited, 8th edition, 2011.
- J. C. Hull and A. White. Incorporating Volatility Updating into the Historical Simulation Method for Value-at-Risk. *Journal of Risk*, 1(1):5–19, 1998.
- S.-H. Hung, C.-H. Lin, and J.-S. Hong. Web Mining for Event-Based Commonsense Knowledge Using Lexico-Syntactic Pattern Matching and Semantic Role Labeling. *Expert Systems with Applications*, 37(1):341–347, 2010.
- W. IJntema, F. Goossen, F. Frasincar, and **F. Hogenboom**. Ontology-Based News Recommendation. In F. Daniel, L. M. L. Delcambre, F. Fotouhi, I. Garrigós, G. Guerrini, J.-N. Mazón, M. Mesiti, S. Müller-Feuerstein, J. Trujillo, T. M. Truta, B. Volz, E. Waller, L. Xiong, and E. Zimányi, editors, *International Workshop on Business intelligencE and the WEB (BEWEB 2010) at 13th International Conference on Extending Database Technology and 13th International Conference on Database Theory (EDBT/ICDT 2010)*, volume 426 of *ACM International Conference Proceeding Series*. ACM, 2010.
- W. IJntema, J. Sangers, **F. Hogenboom**, and F. Frasincar. A Lexico-Semantic Pattern Language for Learning Ontology Instances from Text. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 15(1):37–50, 2012.
- W. IJntema, **F. Hogenboom**, F. Frasincar, and D. Vandic. A Genetic Programming Approach for Learning Semantic Information Extraction Rules from News. In B. Bejnaballah, A. Bestavros, Y. Manolopoulos, A. Vakali, and Y. Zhang, editors, *15th International Conference on Web Information System Engineering (WISE 2014), Part I*, volume 8786 of *Lecture Notes in Computer Science*, pages 418–432. Springer, 2014.
- D. L. Ikenberry and S. Ramnath. Underreaction to Self-Selected News Events: The Case of Stock Splits. *Review of Financial Studies*, 15(2):489–526, 2002.
- P. G. Ipeirotis, F. Provost, and J. Wang. Quality Management on Amazon Mechanical Turk. In *2nd Workshop on Human Computation (HCOMP 2010) at 16th ACM SIGKDD*

- Conference on Knowledge Discovery and Data Mining (KDD 2010)*, pages 64–67. ACM, 2010.
- P. S. Jacobs, G. R. Krupka, and L. F. Rau. Lexico-Semantic Pattern Matching as a Companion to Parsing in Text Understanding. In *Workshop on Speech and Natural Language colocated with the 6th Human Language Technology Conference (HLT 1991)*, pages 337–341. Morgan Kaufmann, 1991.
- A. Java, T. Finin, and S. Nirenburg. SemNews: A Semantic News Framework. In *21st National Conference on Artificial Intelligence (AAAI 2006)*, pages 1939–1940, February 2006.
- J. J. Jiang and D. W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *10th International Conference on Research in Computational Linguistics (ROCLING X)*, pages 19–33, 1997.
- T. Jones. Crossover Macromutation and Population-based Search. In L. J. Eshelman, editor, *6th International Conference on Genetic Algorithms (ICGA 1995)*, pages 73–80. Morgan Kaufmann, 1995.
- P. Jorion. *Value at Risk: The New Benchmark for Managing Financial Risk*. McGraw-Hill, 3rd edition, 2006.
- F. Jungermann and K. Morik. Enhanced Services for Targeted Information Retrieval by Event Extraction and Data Mining. In E. Kapetanios, V. Sugumaran, and M. Spiliopoulou, editors, *13th International Conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems (NLDB 2008)*, volume 5039 of *Lecture Notes in Computer Science*, pages 335–336. Springer Berlin Heidelberg, 2008.
- P. S. Kalev, W.-M. Liu, P. K. Pham, and E. Jarnebic. Public Information Arrival and Volatility of Intraday Stock Returns. *Journal of Banking & Finance*, 28(6):1441–1467, 2004.
- S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic Monitoring and Accident Detection at Intersections. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):108–118, 2000.
- M. J. Kearns and L. E. Ortiz. The Penn-Lehman Automated Trading Project. *IEEE Intelligent Systems*, 18(6):22–31, 2003.

- C. K. Kengne, L. C. Fopa, A. Termier, N. Ibrahim, M.-C. Rousset, T. Washio, and M. Santana. Efficiently Rewriting Large Multimedia Application Execution Traces with Few Event Sequences. In I. S. Dhillon, Y. Koren, R. Ghani, T. E. Senator, P. Bradley, R. Parekh, J. He, R. L. Grossman, and R. Uthurusamy, editors, *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2013)*, pages 1348–1356. ACM, 2013.
- A. J. Keown and J. M. Pinkerton. Merger Announcements and Insider Trading Activity: an Empirical Investigation. *Journal of Finance*, 36(4):855–869, 1981.
- R. Kern, H. Thies, C. Bauer, and G. Satzger. Quality Assurance for Human-Based Electronic Services: A Decision Matrix for Choosing the Right Approach. In F. Daniel and F. M. Facca, editors, *Current Trends in Web Engineering*, volume 6385 of *Lecture Notes in Computer Science*, pages 421–424. Springer-Verlag Berlin Heidelberg, 2010.
- S. Kim, H. Alani, W. Hall, P. H. Lewis, D. E. Millard, N. R. Shadbolt, and M. J. Weal. Artequakt: Generating Tailored Biographies from Automatically Annotated Fragments from the Web. In *Workshop on Semantic Authoring, Annotation & Knowledge Markup (SAAKM 2002)*, pages 1–6, 2002.
- S. T. Kim, J.-C. Lin, and M. B. Slovin. Market Structure, Informed Trading, and Analysts' Recommendations. *Journal of Financial and Quantitative Analysis*, 32(4):507–524, 1997.
- G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax - W3C Recommendation 10 February 2004, 2004. From: <http://www.w3.org/TR/rdf-concepts/>.
- J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- G. Krupka, P. Jacobs, L. Rau, L. Childs, and I. Sider. GE NLToolset: Description of the System as Used for MUC-4. In *4th conference on Message Understanding (MUC 1992)*, pages 177–185. Association for Computational Linguistics, 1992.
- P. H. Kupiec. Techniques for Verifying the Accuracy of Risk Measurement Models. *Journal of Derivatives*, 3(2):73–84, 1995.
- N. Lagos, F. Segond, S. Castellani, and J. O'Neill. Event Extraction for Legal Case Building and Reasoning. In Z. Shi, S. Vadera, A. Aamodt, and D. Leake, editors,

- Intelligent Information Processing V*, volume 340 of *IFIP Advances in Information and Communication Technology*, pages 92–101. Springer Berlin Heidelberg, 2010.
- T. Lauricella, C. S. Stewart, and S. Ovide. Twitter Hoax Sparks Swift Stock Swoon. *The Wall Street Journal*, 2013. From: <http://on.wsj.com/11M3La9>.
- A. Laux and L. Martin. XUpdate, 2000. From: <http://xmldb-org.sourceforge.net/xupdate/xupdate-wd.html>.
- C.-S. Lee, Y.-J. Chen, and Z.-W. Jian. Ontology-Based Fuzzy Event Extraction Agent for Chinese E-News Summarization. *Expert Systems with Applications*, 25(3):431–447, 2003.
- Z. Lei, L.-D. Wu, Y. Zhang, and Y.-C. Liu. A System for Detecting and Tracking Internet News Event. In Y.-S. Ho and H. J. Kim, editors, *6th Pacific-Rim Conference on Multimedia (PCM 2005)*, volume 3767 of *Lecture Notes in Computer Science*, pages 754–764. Springer Berlin Heidelberg, 2005.
- W. Leigh, R. Purvis, and J. M. Ragusa. Forecasting the NYSE Composite Index with Technical Analysis, Pattern Recognizer, Neural Network, and Genetic Algorithm: A Case Study in Romantic Decision Support. *Decision Support Systems*, 32(4):361–377, 2002.
- F. Li, H. Sheng, and D. Zhang. Event Pattern Discovery from the Stock Market Bulletin. In S. Lange, K. Satoh, and C. H. Smith, editors, *5th International Conference on Discovery Science (DS 2002)*, volume 2534 of *Lecture Notes in Computer Science*, pages 35–49. Springer Berlin Heidelberg, 2002.
- D. Lin. An Information-Theoretic Definition of Similarity. In *15th International Conference on Machine Learning (ICML 1998)*, pages 296–304. Morgan Kaufmann, 1998.
- M. Liu, Y. Liu, L. Xiang, X. Chen, and Q. Yang. Extracting Key Entities and Significant Events from Online Daily News. In C. Fyfe, D. Kim, S.-Y. Lee, and H. Yin, editors, *9th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2008)*, volume 5326 of *Lecture Notes in Computer Science*, pages 201–209. Springer Berlin Heidelberg, 2008.
- U. Lösch, S. Rudolph, D. Vrandecic, and R. Studer. Tempus Fugit. In L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, and E. P. B. Simperl, editors, *6th European Semantic Web Conference*

- (ESWC 2009), volume 5554 of *Lecture Notes in Computer Science*, pages 278–292. Springer Berlin Heidelberg, 2009.
- A. G. Maguitman, F. Menczer, H. Roinestad, and A. Vespiagnani. Algorithmic Detection of Semantic Similarity. In A. Ellis and T. Hagino, editors, *14th International Conference on the World Wide Web (WWW 2005)*, pages 107–116. ACM, 2005.
- L. Mancini and F. Trojani. Robust Value at Risk Prediction. *Journal of Financial Econometrics*, 9(2):281–313, 2011.
- C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1st edition, 1999.
- D. Manov, A. Kiryakov, B. Popov, K. Bontcheva, D. Maynard, and H. Cunningham. Experiments with Geographic Knowledge for Information Extraction. In *Workshop on Analysis of Geographic References at 1st Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAAACL 2003)*, pages 1–9. Association for Computational Linguistics, 2003.
- D. Maynard, V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva, and Y. Wilks. Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(1):257–274, 2002.
- D. Maynard, H. Saggion, M. Yankova, K. Bontcheva, and W. Peters. Natural Language Technology for Information Integration in Business Intelligence. In W. Abramowicz, editor, *10th International Conference on Business Information Systems (BIZ 2007)*, volume 4439 of *Lecture Notes in Computer Science*, pages 366–380. Springer Berlin Heidelberg, 2007.
- K. Mehta and S. Bhattacharyya. Adequacy of Training Data for Evolutionary Mining of Trading Rules. *Decision Support Systems*, 37(4):461–474, 2004.
- K. Meijer, F. Frasincar, and **F. Hogenboom**. A Semantic Approach for Extracting Domain Taxonomies from Text. *Decision Support Systems*, 62:78–93, 2014.
- R. Michaely, R. H. Thaler, and K. L. Womack. Price Reactions to Dividend Initiations and Omissions: Overreaction or Drift. *Journal of Finance*, 50(2):573–608, 1995.
- R. Mihalcea and A. Csomai. SenseLearner: Word Sense Disambiguation for All Words in Unrestricted Text. In *43th Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 53–56. Association for Computational Linguistics, 2005.

- A. Mikroyannidis, A. Mantes, and C. Tsalidis. Information Management: The Parmenides Approach. In B. Theodoulidis and C. Tsalidis, editors, *International Workshop on Text Mining Research, Practice and Opportunities at 2nd International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, pages 23–31, 2005.
- V. Milea, F. Frasincar, and U. Kaymak. Knowledge Engineering in a Temporal Semantic Web Context. In *8th International Conference on Web Engineering (ICWE 2008)*, pages 65–74. IEEE Computer Society, 2008.
- V. Milea, F. Frasincar, and U. Kaymak. tOWL: a temporal web ontology language. *IEEE Transactions on Systems, Man and Cybernetics, Part B, Cybernetics*, 42(1):268–281, 2012a.
- V. Milea, F. Frasincar, U. Kaymak, and G.-J. Houben. Temporal Optimizations and Temporal Cardinality in the tOWL Language. *International Journal of Web Engineering and Technology*, 7(1):45–64, 2012b.
- G. Miller, M. Chodorow, S. Landes, C. Leacock, and R. Thomas. Using a Semantic Concordance for Sense Identification. In *Human Language Technology Workshop (HLT 1994)*, pages 240–243. Association for Computational Linguistics, 1994.
- M. L. Mitchell and J. H. Mulherin. The Impact of Public Information on the Stock Market. *Journal of Finance*, 49(3):923–950, 1994.
- M.-A. Mittermayer and G. F. Knolmayer. Text Mining Systems for Market Response to News: A Survey. Technical report, Institute of Information Systems University of Bern, 2006. From: <http://www.ie.iwi.unibe.ch/publikationen/berichte/resource/WP-184>.
- M. Miwa, R. Sætre, J.-D. Kim, and J. Tsujii. Event Extraction With Complex Event Classification Using Rich Features. *Journal of Bioinformatics and Computational Biology*, 8(1):131–146, 2010.
- M. Moerland, **F. Hogenboom**, M. Capelle, and F. Frasincar. Semantics-Based News Recommendation with SF-IDF+. In D. Camacho, R. Akerkar, and M. D. Rodríguez-Moreno, editors, *3rd International Conference on Web Intelligence, Mining and Semantics (WIMS 2013)*. ACM, 2013.
- M. Naughton, N. Kushmerick, and J. Carthy. Event Extraction from Heterogeneous News Sources. In *2006 AAAI Workshop on Event Extraction and Synthesis (W8) at*

- 21st National Conference on Artificial Intelligence (AAAI 2006). From: <http://www.aaai.org/Papers/Workshops/2006/WS-06-07/WS06-07-002.pdf>, 2006.
- R. Navigli. Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2), 2009.
- R. Navigli and P. Velardi. Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1075–1086, 2005.
- S. Nirenburg and V. Raskin. Ontological Semantics, Formal Ontology, and Ambiguity. In *International Conference on Formal Ontology in Information Systems (FOIS 2001)*, pages 151–161. ACM, 2001.
- S. Nirenburg, S. Beale, and M. McShane. Baseline Evaluation of WSD and Semantic Dependency in OntoSem. In J. Bos and R. Delmonte, editors, *1st STEP workshop on Semantics in Text Processing (STEP 2008)*, volume 1 of *Research in Computational Semantics*, pages 179–192. College Publications, 2008.
- Y. Nishihara, K. Sato, and W. Sunayama. Event Extraction and Visualization for Obtaining Personal Experiences from Blogs. In G. Salvendy and M. J. Smith, editors, *Symposium on Human Interface 2009 on Human Interface and the Management of Information. Information and Interaction. Part II*, volume 5618 of *Lecture Notes in Computer Science*, pages 315–324. Springer Berlin Heidelberg, 2009.
- W. Nuij, V. Milea, **F. Hogenboom**, F. Frasincar, and U. Kaymak. An Automated Framework for Incorporating News into Stock Trading Strategies. *IEEE Transactions on Knowledge and Data Engineering*, 26(4):823–835, 2014.
- M. Okamoto and M. Kikuchi. Discovering Volatile Events in Your Neighborhood: Local-Area Topic Extraction from Blog Entries. In G. G. Lee, D. Song, C.-Y. Lin, A. Aizawa, K. Kuriyama, M. Yoshioka, and T. Sakai, editors, *5th Asia Information Retrieval Symposium (AIRS 2009)*, volume 5839 of *Lecture Notes in Computer Science*, pages 181–192. Springer Berlin Heidelberg, 2009.
- D. L. Olson and D. Wu. Value at Risk. In *Enterprise Risk Management Models*, chapter 10, pages 131–141. Springer, 2010.
- C. Pérignon and D. R. Smith. The Level and Quality of Value-at-Risk Disclosure by Commercial Banks. *Journal of Banking & Finance*, 34(2):362–377, 2010.

- M. Phillips. Nasdaq: Here's Our Timeline of the Flash Crash. *The Wall Street Journal*, 2010. From: <http://blogs.wsj.com/marketbeat/2010/05/11/nasdaq-here-s-our-timeline-of-the-flash-crash/>.
- J. Piskorski, H. Tanev, and P. O. Wennerberg. Extracting Violent Events From On-Line News for Ontology Population. In W. Abramowicz, editor, *10th International Conference on Business Information Systems (BIS 2007)*, volume 4439 of *Lecture Notes in Computer Science*, pages 287–300. Springer Berlin Heidelberg, 2007.
- B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. KIM - Semantic Annotation Platform. In Dieter Fensel and Katia Sycara and John Mylopoulos, editor, *2nd International Semantic Web Conference (ISWC 2003)*, volume 2870 of *Lecture Notes in Computer Science*, pages 834–849. Springer Berlin Heidelberg, 2003.
- B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov. KIM – A Semantic Platform For Information Extraction and Retrieval. *Journal of Natural Language Engineering*, 10(3–4):375–392, September 2004a.
- B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov. KIM - A Semantic Platform for Information Extraction and Retrieval. *Journal of Natural Language Engineering*, 10(3–4):375–392, 2004b.
- R. C. Prim. Shortest Connection Networks and Some Generalisations. *Bell System Technical Journal*, 36:1389–1401, 1957.
- E. Prud'hommeaux and A. Seaborne. SPARQL – W3C Recommendation 15 January 2008, 2008. From: <http://www.w3.org/TR/rdf-sparql-query/>.
- K. R. Rampal. The Collection and Flow of World News. In J. C. Merrill, editor, *Global Journalism: Survey of International Communication*, pages 35–52. Longman, 1995.
- I. Ray and I. Ray. Detecting Termination of Active Database Rules Using Symbolic Model Checking. In A. Caplinskas and J. Eder, editors, *5th East European Conference on Advances in Databases and Information Systems (ADBIS 2001)*, volume 2151 of *Lecture Notes in Computer Science*, pages 266–279. Springer Berlin Heidelberg, 2001.
- P. Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, pages 448–453, 1995.

- S. Riedel, H.-W. Chun, T. Takagi, and J. Tsujii. A Markov Logic Approach to Bio-Molecular Event Extraction. In *Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task (BioNLP 2009)*, pages 41–49. Association of Computational Linguistics, 2009.
- F. Rinaldi, J. Dowdall, M. Hess, J. Ellman, G. P. Zarri, A. Persidis, L. Bernard, and H. Karanikas. Multilayer annotations in Parmenides. In *Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2003)*, pages 33–40, 2003.
- F. Rinaldi, G. Schneider, K. Kaljurand, J. Dowdall, C. Andronis, A. Persidis, and O. Konstanti. Mining Relations in the GENIA Corpus. In T. Scheffer, editor, *2nd European Workshop on Data Mining and Text Mining for Bioinformatics at 15th European Conference on Machine Learning (ECML 2004) and the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2004)*, pages 61–68, 2004.
- R. H. Robins. *General Linguistics*. Longman, 4th edition, 1989.
- R. T. Rockafellar and S. Uryasev. Optimization of Conditional Value-at-Risk. *Journal of Risk*, 2(3):21–41, 2000.
- R. T. Rockafellar and S. Uryasev. Conditional Value-at-Risk for General Loss Distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002.
- R. J. Rosen. Merger Momentum and Investor Sentiment: The Stock Market Reaction to Merger Announcements. *Journal of Business*, 79(2):987–1017, 2006.
- B. Rosenberg, K. Reid, and R. Lanstein. Persuasive Evidence of Market Inefficiency. *Journal of Portfolio Management*, 11(3):9–16, 1985.
- H. Saggion, A. Funk, D. Maynard, and K. Bontcheva. Ontology-Based Information Extraction for Business Intelligence. In K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference (ISWC 2007), 2nd Asian Semantic Web Conference (ASWC 2007)*, volume 4825 of *Lecture Notes in Computer Science*, pages 843–856. Springer Berlin Heidelberg, 2007.
- J. Sangers, F. Frasincar, **F. Hogenboom**, A. Hogenboom, and V. Chepegin. A Linguistic Approach for Semantic Web Service Discovery. In J. Casillas, F. J. Martínez-López,

- and J. M. Corchado, editors, *1st International Symposium on Management Intelligent Systems (IS-MiS 2012)*, volume 171 of *Advances in Intelligent Systems and Computing*, pages 131–142. Springer, 2012a.
- J. Sangers, **F. Hogenboom**, and F. Frasincar. Event-Driven Ontology Updating. In X. S. Wang, I. F. Cruz, A. Delis, and G. Huang, editors, *13th International Conference on Web Information System Engineering (WISE 2012)*, volume 7651 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2012b.
- J. Sangers, F. Frasincar, **F. Hogenboom**, and V. Chepegin. Semantic Web Service Discovery Using Natural Language Processing Techniques. *Expert Systems With Applications*, 40(11):4660–4671, 2013.
- K. Schouten, P. Ruijgrok, J. Borsje, F. Frasincar, L. Levering, and **F. Hogenboom**. A Semantic Web-Based Approach for Personalizing News. In W. Chu, W. E. Wong, M. J. Palakal, and C. Hung, editors, *25th Symposium On Applied Computing (SAC 2010), Web Technologies Track*, pages 854–861. ACM, 2010.
- A. Seaborne, G. Manjunath, C. Bizer, J. Breslin, S. Das, I. Davis, S. Harris, K. Idehen, O. Corby, K. Kjernsmo, and B. Nowack. SPARQL Update – W3C Member Submission 15 July 2008, 2008. From: <http://www.w3.org/Submission/SPARQL-Update/>.
- Semlab. ViewerPro. From: <http://www.semlab.nl/portfolio-item/viewerpro-semantic-text-analysis/>, 2013.
- R. Snow, D. Jurafsky, and A. Y. Ng. Learning Syntactic Patterns for Automatic Hypernym Discovery. In *18th Annual Conference on Neural Information Processing Systems (NIPS 2004)*, volume 17 of *Advances in Neural Information Processing Systems*, pages 1297–1304. MIT Press, 2004.
- S. Soderland. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning*, 34(1–3):233–272, 1999.
- A. Street. On the Conditional Value-at-Risk Probability-Dependent Utility Function. *Theory and Decision*, 68(1–2):49–68, 2010.
- Sun Microsystems. Java Compiler Compiler (JavaCC) – The Java Parser Generator, 2013. From: <https://javacc.java.net/>.
- F. G. Tadesse, J. Tekli, R. Chbeir, M. Viviani, and K. Yetongnon. Semantic-based Merging of RSS Items. *World Wide Web Journal, Internet and Web Information Systems, Special Issue on Human-Centered Web Science*, 13(1–2):169–207, 2009.

- R. K. Taira and S. G. Soderland. A Statistical Natural Language Processor for Medical Reports. In *Annual Fall Symposium of the American Medical Informatics Association (AMIA 1999)*, pages 970–974. American Medical Informatics Association, 1999.
- H. Tanev, J. Piskorski, and M. Atkinson. Real-Time News Event Extraction for Global Crisis Monitoring. In E. Kapetanios, V. Sugumaran, and M. Spiliopoulou, editors, *13th International Conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems (NLDB 2008)*, volume 5039 of *Lecture Notes in Computer Science*, pages 207–218. Springer Berlin Heidelberg, 2008.
- P. C. Tetlock. Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *Journal of Finance*, 62(3):1139–1168, 2007.
- P. C. Tetlock. More than Words: Quantifying Language to Measure Firms' Fundamentals. *Journal of Finance*, 63(3):1437–1467, 2008.
- The Apache Software Foundation. Apache Jena – A Free and Open Source Java Framework for Building Semantic Web and Linked Data Applications, 2013. From: <http://jena.apache.org/>.
- The Economist. Ahead of the Tape: The Best Newsreaders May Soon Be Computers. *The Economist*, 383:85, 2007.
- M. Theobald, R. Schenkel, and G. Weikum. Exploiting Structure, Annotation, and Ontological Knowledge for Automatic Classification of XML Data. In V. Christopoulos and J. Freire, editors, *6th International Workshop on the Web and Databases (WebDB 2003)*, pages 1–6. IEEE Computer Society, 2003.
- D. R. Thompson and G. L. Bilbro. Comparison of a Genetic Algorithm with a Simulated Annealing Algorithm for the Design of an ATM Network. *IEEE Communications Letters*, 4(8):267–269, 2000.
- M.-V. Tran, M.-H. Nguyen, S.-Q. Nguyen, M.-T. Nguyen, and X.-H. Phan. VnLoc: A Real – Time News Event Extraction Framework for Vietnamese. In *4th International Conference on Knowledge and Systems Engineering (KSE 2012)*, pages 161–166. IEEE Computer Society, 2012.
- J. van Bommel. Rumors. *Journal of Finance*, 58(4):1499–1520, 2003.
- S. van Landeghem, J. Björne, C.-H. Wei, K. Hakala, S. Pyysalo, S. Ananiadou, H.-Y. Kao, Z. Lu, T. Salakoski, Y. van de Peer, and F. Ginter. Large-Scale Event Extraction from Literature with Multi-Level Gene Normalization. *PLoS One*, 8(4):e55814, 2013.

- C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 2nd edition, 1979.
- D. Vandic, L. J. Nederstigt, S. S. Aanen, F. Frasincar, and **F. Hogenboom**. Ontology Population from Web Product Information. In C.-W. Chung, A. Z. Broder, K. Shim, and T. Suel, editors, *23rd International World Wide Web Conference (WWW Companion 2014)*, pages 391–392. ACM, 2014.
- M. Vargas-Vera and D. Celjuska. Event Recognition on News Stories and Semi-Automatic Population of an Ontology. In *3rd IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004)*, pages 615–618. IEEE Computer Society, 2004.
- A. Verheij, A. Kleijn, F. Frasincar, and **F. Hogenboom**. A Comparison Study for Novelty Control Mechanisms Applied to Web News Stories. In N. Zhong, Z. Gong, Y. ming Cheung, P. Lingras, P. S. Szczepaniak, and E. Suzuki, editors, *The 2012 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2012)*, pages 431–436. IEEE Computer Society, 2012a.
- A. Verheij, A. Kleijn, F. Frasincar, D. Vandic, and **F. Hogenboom**. Supporting the Negation Operator in the Hermes Graphical Query Language for Document Ranking. In T. Demeester, J. Deleu, L. Mertens, D. Plaetinck, A. de Moor, T. Hoang, T. Wauters, C. Develder, B. Vermeulen, and P. Demeester, editors, *12th Dutch-Belgian Information Retrieval Workshop (DIR 2012)*, pages 73–74, 2012b.
- A. Verheij, A. Kleijn, F. Frasincar, D. Vandic, and **F. Hogenboom**. Querying and Ranking News Items in the Hermes Framework. In S. Ossowski and P. Lecca, editors, *27th Symposium on Applied Computing (SAC 2012), Web Technologies Track*, pages 672–679. ACM, 2012c.
- J. B. Warner, R. L. Watts, and K. H. Wruck. Stock Prices and Top Management Changes. *Journal of Financial Economics*, 20(1):461–492, 1988.
- C.-P. Wei and Y.-H. Lee. Event detection from Online News Documents for Supporting Environmental Scanning. *Decision Support Systems*, 36(4):385–401, 2004.
- P. Wichmann, A. Borek, R. Kern, P. Woodal, A. K. Parlikad, and G. Satzger. Exploring the “Crowd” as Enabler of Better Information Quality. In A. Koronios and J. Gao, editors, *16th International Conference on Information Quality (ICIQ 2011)*, pages 302–312. Curran Associates, Inc., 2011.
- D. Winer. RSS 2.0 Specification, 2003. From: <http://cyber.law.harvard.edu/rss/rss.html>.

- F. Xu, H. Uszkoreit, and H. Li. Automatic Event and Relation Detection with Seeds of Varying Complexity. In *2006 AAAI Workshop on Event Extraction and Synthesis (W8) at 21st National Conference on Artificial Intelligence (AAAI 2006)*. From: <http://www.aaai.org/Papers/Workshops/2006/WS-06-07/WS06-07-004.pdf>, 2006.
- A. Yakushiji, Y. Tateisi, and Y. Miyao. Event Extraction from Biomedical Papers using a Full Parser. In *6th Pacific Symposium on Biocomputing (PSB 2001)*, pages 408–419, 2001.
- D. Yuret. Some experiments with a Naive Bayes WSD System. In R. Mihalcea and P. Edmonds, editors, *3rd ACL/SIGLEX International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (senseval-3)*, pages 265–268. Association for Computational Linguistics, 2004.
- G. P. Zarri. NKRL, a Knowledge Representation Tool for Encoding the ‘Meaning’ of Complex Narrative Texts. *Natural Language Engineering*, 3(2):231–253, 1997.
- Y. Zhai, A. Hsu, and S. K. Halgamuge. Combining News and Technical Indicators in Daily Stock Price Trends Prediction. In D. Liu, S. Fei, Z. Hou, H. Zhang, and C. Sun, editors, *4th International Symposium on Neural Networks (ISSN 2007)*, pages 1087–1096. Springer-Verlag Berlin, Heidelberg, 2007.
- X. F. Zhang. Information Uncertainty and Stock Returns. *Journal of Finance*, 61(1):105–137, 2006.
- H. Zhao. A Multi-Objective Genetic Programming Approach to Developing Pareto Optimal Decision Trees. *Decision Support Systems*, 43(3):809–826, 2007.

Summary in English

The flourishing data market and the financial resources pumped into research on the extraction of knowledge from data underline today's major stakes that are involved with the accurate extraction of knowledge and efficient usage thereof in financial decision making. Not only traditional and ubiquitous numerical data, but especially textual data such as news messages, are gradually receiving more attention. Such data often remain unused, due to their unstructured nature thwarting automatic processing. Thus, often large amounts of valuable knowledge remain undiscovered. Generally, this knowledge can be summarized in events, which are inextricably linked with financial markets. It has long been known that events, like acquisitions, product launches, natural disasters, or wars, could exert a notable influence on stock rates. Smart applications of detected events may therefore be beneficial for financial decision making. For instance, trading algorithms can be enriched with events, so as to better respond to new developments. Also, taking into account certain events may improve financial risk estimations. Because of the promising applications, but also the many inevitable challenges, this dissertation comprises a number of topics related to the semi-automatic detection of (financial) events in news text.

Taking into consideration the results of a detailed evaluation of currently existing knowledge-driven, data-driven, and hybrid extraction systems and methods focusing on events, this thesis presents a semi-automatic system for financial event extraction from news text. The system consists of various innovative, qualitatively competitive, and knowledge-driven components for natural language processing. Their interaction with a knowledge base fosters a feedback loop, so that newly detected events can be digested, allowing the incorporated knowledge to be used in future (extraction) processes.

In addition, two languages are proposed that can refine the aforementioned system. While the first language is focused on pattern-based event extraction from text, the second language is targeted toward the definition and execution of knowledge base updates, associated with the extracted events. After conducting a series of experiments focusing on pattern development times and result accuracies, it can be concluded that the

extraction language offers, in contrast to many modern alternatives, a simple and flexible notation utilizing lexical, syntactical, and semantical elements, while maintaining expressivity. Moreover, an evolutionary approach for automatically generating patterns contributes to the practical employability of the language. The trigger-based knowledge base update language and various developed execution models are particularly suitable for event extraction applications. While modern languages are often less suited for fully automated updates, now an increased flexibility is offered for automatically executing pre-defined rules, providing the user with a wide range of options, e.g., immediate or deferred execution, update chaining, etcetera.

Last, two financial applications of events extracted from news text are presented in this dissertation. For both applications it is confirmed that the addition of events to prevailing computations or algorithms can yield more accurate results. In our analysis of an event-based automated trading application, rules are generated for trading stocks. The best performing rules do not only make use of numerical signals such as average historical stock rates, but also employ news-based event signals. Moreover, when cleaning stock data from disruptions caused by financial events, financial risk analyses yield more accurate results.

The reported results suggest that it is possible to semi-automatically and accurately detect events in news text in a knowledge-driven way, when making use of advanced extraction rules. Additional update rules and execution models accommodate a feedback loop to knowledge bases underlying event extraction systems. Events detected in news can be used as additional parameters in financial applications, thus yielding more accurate outcomes in the evaluated cases. Such advantageous applications can be of good use in (semi-)automated environments. Given these considerations, we envision a bright future for event detection.

Nederlandse Samenvatting (Summary in Dutch)

De florerende datamarkt en de financiële middelen die worden gepompt in onderzoek naar de extractie van kennis uit data, onderstrepen de grote belangen die vandaag de dag gemoeid zijn met het accuraat destilleren van kennis en het efficiënt gebruiken hiervan in financiële beslissingsprocessen. Niet alleen voor de traditionele en alomtegenwoordige numerieke data, maar juist voor tekstuele data als nieuwsberichten, groeit de interesse gestaag. Dergelijke data blijft vaak ongebruikt, omdat het ongestructureerd en dus moeilijk (automatisch) te verwerken is, maar bevat daarentegen wél veel waardevolle kennis. Deze kennis kan men doorgaans samenvatten in gebeurtenissen, welke onlosmakelijk verbonden zijn met financiële markten. Het is al langer bekend dat gebeurtenissen, zoals overnames en productlanceringen, maar bijvoorbeeld ook natuur rampen en oorlogen, hun sporen kunnen nalaten in aandelenkoersen. Het slim toepassen van gedetecteerde gebeurtenissen kan dus voordelig zijn bij het nemen van financiële beslissingen. Handelsalgoritmes kunnen bijvoorbeeld verrijkt worden met gebeurtenissen, om zo beter in te spelen op nieuwe ontwikkelingen. Tevens kunnen financiële risico's beter geschat worden, wanneer rekening gehouden wordt met bepaalde gebeurtenissen. Gezien de veelbelovende toepassingen, maar ook de vele onvermijdelijke uitdagingen, behandelt deze dissertatie enkele onderzoeken omtrent de semi-automatische detectie van (financiële) gebeurtenissen uit nieuwsberichten.

Naar de resultaten van een gedetailleerde evaluatie van reeds bestaande kennisgedreven, datagedreven en hybride extractiesystemen en -methoden die zich toleggen op gebeurtenissen, wordt in dit proefschrift een semi-automatisch systeem ten behoeve van de extractie van financiële gebeurtenissen uit nieuwsberichten gepresenteerd. Dit systeem bestaat uit enkele vernieuwende, kwalitatief competitieve, kennisgedreven componenten voor de verwerking van tekst. Door hun interactie met een kennisbank wordt een terugkoppeling mogelijk gemaakt, zodat nieuw gedetecteerde gebeurtenissen verwerkt, en tevens in toekomstige (extractie)processen toegepast kunnen worden.

Daarnaast staan twee ontwikkelde talen centraal, die het voorgenoemde systeem kunnen verfijnen. Waar de eerste taal zich richt op het extraheren van gebeurtenissen uit tekst met behulp van patronen, richt de tweede taal zich juist op het definiëren en uitvoeren van de met de gedetecteerde gebeurtenissen geassocieerde kennisbankwijzigingen. Na een onderwerping aan een serie experimenten gericht op de ontwikkeltijden van de patronen en de nauwkeurigheid van de resultaten, kan geconcludeerd worden dat de extractietaal, in tegenstelling tot veel hedendaagse alternatieven, een eenvoudige en flexibele notatie biedt waarbij gebruik gemaakt kan worden van lexicale, syntactische en semantische elementen, terwijl de expressiviteit gewaarborgd wordt. Een evolutionaire aanpak om automatisch dergelijke patronen te genereren draagt bovendien bij aan de praktische inzetbaarheid van de taal. De in de dissertatie nader beschreven taal en enkele ontwikkelde uitvoeringsmodellen voor het automatisch bijwerken van een kennisbank aan de hand van signalen, is uitermate geschikt om te gebruiken in de context van gebeurtenissenextractie. Waar huidige talen zich vaak minder lenen voor volledig geautomatiseerde bewerkingen, wordt nu meer flexibiliteit geboden voor het automatisch uitvoeren van vooraf gedefinieerde regels, en heeft de gebruiker bijvoorbeeld de keuze uit het direct of juist uitgesteld uit laten voeren van bewerkingen, het aaneen laten schakelen van aanpassingen, enzovoorts.

Tot slot worden in dit proefschrift twee financiële toepassingen van uit nieuwsberichten onttrokken gebeurtenissen uitgelicht. Bij beide toepassingen kan worden vastgesteld dat de toevoeging van gebeurtenissen in gangbare berekeningen of algoritmes kan leiden tot nauwkeurigere resultaten. Zo genereren we in het geval van geautomatiseerde handelsalgoritmes regels voor het verhandelen van aandelen, en tonen we aan dat, wanneer niet alleen numerieke signalen zoals gemiddelde historische koersen, maar ook nieuwssignalen worden verwerkt in dergelijke regels, dit doorgaans leidt tot winstgevende strategieën. Wanneer we daarnaast bij financiële risicoanalyses gebruik maken van data die geschoond is van verstoringen veroorzaakt door financiële gebeurtenissen, blijkt ook hier dat door rekening te houden met dergelijke gebeurtenissen, de resultaten verbeteren.

Uit de gerapporteerde resultaten valt op te maken dat het mogelijk is om op een semi-automatische, kennis-gedreven wijze accuraat gebeurtenissen te detecteren in nieuwsberichten, wanneer gebruik gemaakt wordt van geavanceerde extractieregels. Aanvullende regels en uitvoeringsmodellen maken een terugkoppeling mogelijk naar kennisbanken die ten grondslag liggen aan gebeurtenissenextractiesystemen. In tekst gedetecteerde gebeurtenissen kunnen gebruikt worden als additionele parameters in financiële applicaties, wat in de geteste gevallen leidt tot nauwkeurigere uitkomsten. Deze veelbelovende applicaties kunnen goed van pas komen in (semi-)geautomatiseerde omgevingen. Dit inachtnemend, is voor de detectie van gebeurtenissen een rooskleurige toekomst weggelegd.

About the Author



Frederik Hogenboom (April 13, 1987) obtained cum laude the M.Sc. degree in Economics and Informatics from the Erasmus University Rotterdam, the Netherlands, in 2009, specializing in Computational Economics. Already during his Bachelor's and Master's programmes, he published research that mainly focused on the Semantic Web and learning agents.

Under the auspices of the Erasmus Research Institute of Management (ERIM) and the Econometric Institute at the Erasmus School of Economics, Frederik continued his line of research in a Ph.D. candidacy supported by the Netherlands Organization for Scientific Research (NWO) – under the Physical Sciences Free Competition project 612.001.009: Financial Events Recognition in News for Algorithmic Trading (FERNAT) – and the Dutch national public-private research community program COMMIT, where his work was linked to the Infiniti project. In his years at the Erasmus University Rotterdam, Frederik was additionally affiliated to the Erasmus Center of Business Intelligence (ECBI), Erasmus Studio, and the Dutch Research School for Information and Knowledge Systems (SIKS).

Frederik's current research is primarily targeted toward the multidisciplinary field of business intelligence, where the main focus is on ways to employ financial event discovery in emerging news for algorithmic trading, hereby combining techniques from various disciplines, amongst which Semantic Web, text mining, artificial intelligence, machine learning, linguistics, and finance. Over the years, Frederik has published many papers at prestigious international conferences (e.g., DEXA, ER, ISWC, WI, WISE, and WWW), was active at national conferences like BNAIC, DBDBD, DIR, and ICT.OPEN with numerous contributions, but also actively ventured to other outlets such as Economie Opinie. Moreover, he has written a handful of book chapters and published a substantial amount articles in renowned journals such as DKE, DSS, ESWA, JWS, and TKDE, with an additional contribution still under review.

Frederik has reviewed submissions for many international conferences (e.g., SMC and CIKM) and journals (e.g., DSS, EAAI, and ESWA), and has served as a program committee member in multiple tracks and editions of the international conferences SAC and ICCCI. Also, Frederik was active in the role of session chair (at DEXA and SAC) and local organizer of IFSA/EUSFLAT and DBDBD. Moreover, Frederik was involved in teaching over a dozen programming and other IT-related courses – obtaining consistent outstanding student evaluations – and the supervision of many Bachelor's and Master's theses. Last, he has served multiple years in the University Ph.D. Council (EPAR) during his Ph.D. candidacy, cooperating with departmental and national Ph.D. councils and giving Erasmus Ph.D. candidates a face and voice at the university level.

ERIM Ph.D. Series Overview

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT (ERIM)

ERIM PH.D. SERIES RESEARCH IN MANAGEMENT

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: <http://hdl.handle.net/1765/1>. ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics at the Erasmus University Rotterdam (EUR).

DISSERTATIONS LAST FIVE YEARS

Acciaro, M., *Bundling Strategies in Global Supply Chains*, Promoter(s): Prof.dr. H.E. Haralambides, EPS-2010-197-LIS, <http://hdl.handle.net/1765/19742>

Agatz, N.A.H., *Demand Management in E-Fulfillment*, Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen, EPS-2009-163-LIS, <http://hdl.handle.net/1765/15425>

Akpınar, E., *Consumer Information Sharing*, Promoter(s): Prof.dr.ir. A. Smidts, EPS-2013-297-MKT, <http://hdl.handle.net/1765/50140>

Alexiev, A.S., *Exploratory Innovation: The Role of Organizational and Top Management Team Social Capital*, Promoter(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-208-STR, <http://hdl.handle.net/1765/20632>

Asperen, E. van, *Essays on Port, Container, and Bulk Chemical Logistics Optimization*, Promoter(s): Prof.dr.ir. R. Dekker, EPS-2009-181-LIS, <http://hdl.handle.net/1765/17626>

Ates, M.A., *Purchasing and Supply Management at the Purchase Category Level: strategy, structure and performance*, Promoter(s): Prof.dr. J.Y.F. Wynstra & Dr. E.M. van Raaij, EPS-2014-300-LIS, <http://hdl.handle.net/1765/50283>

Bannouh, K., *Measuring and Forecasting Financial Market Volatility using High-Frequency Data*, Promoter(s): Prof.dr. D.J.C. van Dijk, EPS-2013-273-F&A, <http://hdl.handle.net/1765/38240>

Ben-Menahem, S.M., *Strategic Timing and Proactiveness of Organizations*, Promoter(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2013-278-S&E, <http://hdl.handle.net/1765/39128>

Benning, T.M., *A Consumer Perspective on Flexibility in Health Care: Priority Access Pricing and Customized Care*, Promoter(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2011-241-MKT, <http://hdl.handle.net/1765/23670>

Berg, W.E. van den, *Understanding Salesforce Behavior using Genetic Association Studies*, Promoter(s): Prof.dr. W.J.M.I. Verbeke, EPS-2014-311-MKT, <http://hdl.handle.net/1765/51440>

Betancourt, N.E., *Typical Atypicality: Formal and Informal Institutional Conformity, Deviance, and Dynamics*, Promoter(s): Prof.dr. B. Krug, EPS-2012-262-ORG, <http://hdl.handle.net/1765/32345>

Bezemter, P.J., *Diffusion of Corporate Governance Beliefs: Board independence and the emergence of a shareholder value orientation in the Netherlands*, Promoter(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-192-STR, <http://hdl.handle.net/1765/18458>

Binken, J.L.G., *System markets: Indirect network effects in action, or inaction?*, Promoter(s): Prof.dr. S. Stremersch, EPS-2010-213-MKT, <http://hdl.handle.net/1765/21186>

Blitz, D.C., *Benchmarking Benchmarks*, Promoter(s): Prof.dr. A.G.Z. Kemna & Prof.dr. W.F.C. Verschoor, EPS-2011-225-F&A, <http://hdl.handle.net/1765/22624>

Boons, M., *Working Together Alone in the Online Crowd: The Effects of Social Motivations and Individual Knowledge Backgrounds on the Participation and Performance of Members of Online Crowdsourcing Platforms*, Promoter(s): Prof.dr. H.G. Barkema & Dr. D.A. Stam, EPS-2014-306-S&E, <http://hdl.handle.net/1765/50711>

Borst, W.A.M., *Understanding Crowdsourcing: Effects of motivation and rewards on participation and performance in voluntary online activities*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende & Prof.dr.ir. H.W.G.M van Heck, EPS-2010-221-LIS, <http://hdl.handle.net/1765/21914>

Budiono, D.P., *The Analysis of Mutual Fund Performance: Evidence from U.S. Equity Mutual Funds*, Promoter(s): Prof.dr. M.J.C.M. Verbeek & Dr.ir. M.P.E. Martens, EPS-2010-185-F&A, <http://hdl.handle.net/1765/18126>

Burger, M.J., *Structure and Cooptition in Urban Networks*, Promoter(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.R. Commandeur, EPS-2011-243-ORG, <http://hdl.handle.net/1765/26178>

Byington, E., *Exploring Coworker Relationships: Antecedents and Dimensions of Interpersonal Fit, Coworker Satisfaction, and Relational Models*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2013-292-ORG, <http://hdl.handle.net/1765/41508>

Camacho, N.M., *Health and Marketing: Essays on Physician and Patient Decision-Making*, Promoter(s): Prof.dr. S. Stremersch, EPS-2011-237-MKT, <http://hdl.handle.net/1765/23604>

Caron, E.A.M., *Explanation of Exceptional Values in Multi-dimensional Business Databases*, Promoter(s): Prof.dr. H.A.M. Daniels & Prof.dr. G.W.J. Hendrikse, EPS-2013-296-LIS, <http://hdl.handle.net/1765/50005>

Carvalho, L. de, *Knowledge Locations in Cities: Emergence and Development Dynamics*, Promoter(s): Prof.dr. L. Berg, EPS-2013-274-S&E, <http://hdl.handle.net/1765/38449>

Carvalho de Mesquita Ferreira, L., *Attention Mosaics: Studies of Organizational Attention*, Promoter(s): Prof.dr. P.M.A.R. Heugens & Prof.dr. J. van Oosterhout, EPS-2010-205-ORG, <http://hdl.handle.net/1765/19882>

Chen, C.M., *Evaluation and Design of Supply Chain Operations using DEA*, Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen, EPS-2009-172-LIS, <http://hdl.handle.net/1765/16181>

Cox, R.H.G.M., *To Own, To Finance, and To Insure - Residential Real Estate Revealed*, Promoter(s): Prof.dr. D. Brounen, EPS-2013-290-F&A, <http://hdl.handle.net/1765/40964>

Defilippi Angeldonis, E.F., *Access Regulation for Naturally Monopolistic Port Terminals: Lessons from Regulated Network Industries*, Promoter(s): Prof.dr. H.E. Haralambides, EPS-2010-204-LIS, <http://hdl.handle.net/1765/19881>

Deichmann, D., *Idea Management: Perspectives from Leadership, Learning, and Network Theory*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2012-255-ORG, <http://hdl.handle.net/1765/31174>

Desmet, P.T.M., *In Money we Trust? Trust Repair and the Psychology of Financial Compensations*, Promoter(s): Prof.dr. D. de Cremer, EPS-2011-232-ORG, <http://hdl.handle.net/1765/23268>

Diepen, M. van, *Dynamics and Competition in Charitable Giving*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2009-159-MKT, <http://hdl.handle.net/1765/14526>

Dietvorst, R.C., *Neural Mechanisms Underlying Social Intelligence and Their Relationship with the Performance of Sales Managers*, Promoter(s): Prof.dr. W.J.M.I. Verbeke, EPS-2010-215-MKT, <http://hdl.handle.net/1765/21188>

Dietz, H.M.S., *Managing (Sales)People towards Performance: HR Strategy, Leadership & Teamwork*, Promoter(s): Prof.dr. G.W.J. Hendrikse, EPS-2009-168-ORG, <http://hdl.handle.net/1765/16081>

Dollevoet, T.A.B., *Delay Management and Dispatching in Railways*, Promoter(s): Prof.dr. A.P.M. Wagelmans, EPS-2013-272-LIS, <http://hdl.handle.net/1765/38241>

Doorn, S. van, *Managing Entrepreneurial Orientation*, Promoter(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch, & Prof.dr. H.W. Volberda, EPS-2012-258-STR, <http://hdl.handle.net/1765/32166>

Douwens-Zonneveld, M.G., *Animal Spirits and Extreme Confidence: No Guts, No Glory?*, Promoter(s): Prof.dr. W.F.C. Verschoor, EPS-2012-257-F&A, <http://hdl.handle.net/1765/31914>

Duca, E., *The Impact of Investor Demand on Security Offerings*, Promoter(s): Prof.dr. A. de Jong, EPS-2011-240-F&A, <http://hdl.handle.net/1765/26041>

Duursema, H., *Strategic Leadership: Moving Beyond the Leader-Follower Dyad*, Promoter(s): Prof.dr. R.J.M. van Tulder, EPS-2013-279-ORG, <http://hdl.handle.net/1765/39129>

Eck, N.J. van, *Methodological Advances in Bibliometric Mapping of Science*, Promoter(s): Prof.dr.ir. R. Dekker, EPS-2011-247-LIS, <http://hdl.handle.net/1765/26509>

Eijk, A.R. van der, *Behind Networks: Knowledge Transfer, Favor Exchange and Performance*, Promoter(s): Prof.dr. S.L. van de Velde & Prof.dr. W.A. Dolsma, EPS-2009-161-LIS, <http://hdl.handle.net/1765/14613>

Essen, M. van, *An Institution-Based View of Ownership*, Promoter(s): Prof.dr. J. van Oosterhout & Prof.dr. G.M.H. Mertens, EPS-2011-226-ORG, <http://hdl.handle.net/1765/22643>

Feng, L., *Motivation, Coordination and Cognition in Cooperatives*, Promoter(s): Prof.dr. G.W.J. Hendrikse, EPS-2010-220-ORG, <http://hdl.handle.net/1765/21680>

Gertsen, H.F.M., *Riding a Tiger without Being Eaten: How Companies and Analysts Tame Financial Restatements and Influence Corporate Reputation*, Promoter(s): Prof.dr. C.B.M. van Riel, EPS-2009-171-ORG, <http://hdl.handle.net/1765/16098>

Gharehgozli, A.H., *Developing New Methods for Efficient Container Stacking Operations*, Promoter(s): Prof.dr.ir. M.B.M. de Koster, EPS-2012-269-LIS, <http://hdl.handle.net/1765/37779>

Gijsbers, G.W., *Agricultural Innovation in Asia: Drivers, Paradigms and Performance*, Promoter(s): Prof.dr. R.J.M. van Tulder, EPS-2009-157-ORG, <http://hdl.handle.net/1765/14524>

Gils, S. van, *Morality in Interactions: On the Display of Moral Behavior by Leaders and Employees*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2012-270-ORG, <http://hdl.handle.net/1765/38027>

Ginkel-Bieshaar, M.N.G. van, *The Impact of Abstract versus Concrete Product Communications on Consumer Decision-making Processes*, Promoter(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2012-256-MKT, <http://hdl.handle.net/1765/31913>

Gkougkousi, X., *Empirical Studies in Financial Accounting*, Promoter(s): Prof.dr. G.M.H. Mertens & Prof.dr. E. Peek, EPS-2012-264-F&A, <http://hdl.handle.net/1765/37170>

Gong, Y., *Stochastic Modelling and Analysis of Warehouse Operations*, Promoter(s): Prof.dr. M.B.M. de Koster & Prof.dr. S.L. van de Velde, EPS-2009-180-LIS, <http://hdl.handle.net/1765/16724>

Greeven, M.J., *Innovation in an Uncertain Institutional Environment: Private Software Entrepreneurs in Hangzhou, China*, Promoter(s): Prof.dr. B. Krug, EPS-2009-164-ORG, <http://hdl.handle.net/1765/15426>

Günster, N., *Investment Strategies based on Social Responsibility and Bubbles*, Promoter(s): Prof.dr. C.G. Koedijk, EPS-2009-175-F&A, <http://hdl.handle.net/1765/16209>

Hakimi, N.A., *Leader Empowering Behaviour: The Leader's Perspective*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2010-184-ORG, <http://hdl.handle.net/1765/17701>

Hensmans, M., *A Republican Settlement Theory of the Firm: Applied to Retail Banks in England and the Netherlands (1830-2007)*, Promoter(s): Prof.dr. A. Jolink & Prof.dr. S.J. Magala, EPS-2010-193-ORG, <http://hdl.handle.net/1765/19494>

Hernández-Mireles, C., *Marketing Modeling for New Products*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2010-202-MKT, <http://hdl.handle.net/1765/19878>

Heyde Fernandes, D. von der, *The Functions and Dysfunctions of Reminders*, Promoter(s): Prof.dr. S.M.J. van Osselaer, EPS-2013-295-MKT, <http://hdl.handle.net/1765/41514>

Heyden, M.L.M., *Essays on Upper Echelons & Strategic Renewal: A Multilevel Contingency Approach*, Promoter(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-259-STR, <http://hdl.handle.net/1765/32167>

Hoever, I.J., *Diversity and Creativity*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2012-267-ORG, <http://hdl.handle.net/1765/37392>

Hoogendoorn, B., *Social Entrepreneurship in the Modern Economy: Warm Glow, Cold Feet*, Promoter(s): Prof.dr. H.P.G. Pennings & Prof.dr. A.R. Thurik, EPS-2011-246-STR, <http://hdl.handle.net/1765/26447>

Hoogervorst, N., *On The Psychology of Displaying Ethical Leadership: A Behavioral Ethics Approach*, Promoter(s): Prof.dr. D. de Cremer & Dr. M. van Dijke, EPS-2011-244-ORG, <http://hdl.handle.net/1765/26228>

Hout, D.H. van, *Measuring Meaningful Differences: Sensory Testing Based Decision Making in an Industrial Context; Applications of Signal Detection Theory and Thurstonian*

Modelling, Promoter(s): Prof.dr. P.J.F. Groenen & Prof.dr. G.B. Dijksterhuis, EPS-2014-304-MKT, <http://hdl.handle.net/1765/50387>

Huang, X., *An Analysis of Occupational Pension Provision: From Evaluation to Redesign*, Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2010-196-F&A, <http://hdl.handle.net/1765/19674>

Hytönen, K.A., *Context Effects in Valuation, Judgment and Choice: A Neuroscientific Approach*, Promoter(s): Prof.dr.ir. Ale Smidts, EPS-2011-252-MKT, <http://hdl.handle.net/1765/30668>

Jaarsveld, W.L. van, *Maintenance Centered Service Parts Inventory Control*, Promoter(s): Prof.dr.ir. R. Dekker, EPS-2013-288-LIS, <http://hdl.handle.net/1765/39933>

Jalil, M.N., *Customer Information Driven After Sales Service Management: Lessons from Spare Parts Logistics*, Promoter(s): Prof.dr. L.G. Kroon, EPS-2011-222-LIS, <http://hdl.handle.net/1765/22156>

Jaspers, F.P.H., *Organizing Systemic Innovation*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2009-160-ORG, <http://hdl.handle.net/1765/14974>

Jiang, T., *Capital Structure Determinants and Governance Structure Variety in Franchising*, Promoter(s): Prof.dr. G.W.J. Hendrikse & Prof.dr. A. de Jong, EPS-2009-158-ORG, <http://hdl.handle.net/1765/14975>

Jiao, T., *Essays in Financial Accounting*, Promoter(s): Prof.dr. G.M.H. Mertens, EPS-2009-176-F&A, <http://hdl.handle.net/1765/16097>

Kaa, G. van de, *Standards Battles for Complex Systems: Empirical Research on the Home Network*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-166-LIS, <http://hdl.handle.net/1765/16011>

Kagie, M., *Advances in Online Shopping Interfaces: Product Catalog Maps and Recommender Systems*, Promoter(s): Prof.dr. P.J.F. Groenen, EPS-2010-195-MKT, <http://hdl.handle.net/1765/19532>

Kappe, E.R., *The Effectiveness of Pharmaceutical Marketing*, Promoter(s): Prof.dr. S. Stremersch, EPS-2011-239-MKT, <http://hdl.handle.net/1765/23610>

Karreman, B., *Financial Services and Emerging Markets*, Promoter(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.P.G. Pennings, EPS-2011-223-ORG, <http://hdl.handle.net/1765/22280>

Kil, J., *Acquisitions Through a Behavioral and Real Options Lens*, Promoter(s): Prof.dr. H.T.J. Smit, EPS-2013-298-F&A, <http://hdl.handle.net/1765/50142>

Klooster, E. van 't, *Travel to Learn: the Influence of Cultural Distance on Competence Development in Educational Travel*, Promoter(s): Prof.dr. F.M. Go & Prof.dr. P.J. van Baalen, EPS-2014-312-MKT, <http://hdl.handle.net/1765/514462>

Koendjbiharie, S.R., *The Information-Based View on Business Network Performance: Revealing the Performance of Interorganizational Networks*, Promoter(s): Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. P.H.M. Vervest, EPS-2014-315-LIS, <http://hdl.handle.net/1765/51751>

Konter, D.J., *Crossing Borders with HRM: An Inquiry of the Influence of Contextual Differences in the Adoption and Effectiveness of HRM*, Promoter(s): Prof.dr. J. Paauwe & Dr. L.H. Hoeksema, EPS-2014-305-ORG, <http://hdl.handle.net/1765/50388>

Korkmaz, E., *Bridging Models and Business: Understanding Heterogeneity in Hidden Drivers of Customer Purchase Behavior*, Promoter(s): Prof.dr. S.L. van de Velde & Prof.dr. D. Fok, EPS-2014-316-LIS, <http://hdl.handle.net/1765/76008>

Kwee, Z., *Investigating Three Key Principles of Sustained Strategic Renewal: A Longitudinal Study of Long-Lived Firms*, Promoter(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2009-174-STR, <http://hdl.handle.net/1765/16207>

Lam, K.Y., *Reliability and Rankings*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2011-230-MKT, <http://hdl.handle.net/1765/22977>

Lander, M.W., *Profits or Professionalism? On Designing Professional Service Firms*, Promoter(s): Prof.dr. J. van Oosterhout & Prof.dr. P.P.M.A.R. Heugens, EPS-2012-253-ORG, <http://hdl.handle.net/1765/30682>

Langhe, B. de, *Contingencies: Learning Numerical and Emotional Associations in an Uncertain World*, Promoter(s): Prof.dr.ir. B. Wierenga & Prof.dr. S.M.J. van Osselaer, EPS-2011-236-MKT, <http://hdl.handle.net/1765/23504>

Larco Martinelli, J.A., *Incorporating Worker-Specific Factors in Operations Management Models*, Promoter(s): Prof.dr.ir. J. Dul & Prof.dr. M.B.M. de Koster, EPS-2010-217-LIS, <http://hdl.handle.net/1765/21527>

Leunissen, J.M., *All Apologies: On the Willingness of Perpetrators to Apologize*, Promoter(s): Prof.dr. D. de Cremer & Dr. M. van Dijke, EPS-2014-301-ORG, <http://hdl.handle.net/1765/50318>

Li, T., *Informedness and Customer-Centric Revenue management*, Promoter(s): Prof.dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-158-LIS, <http://hdl.handle.net/1765/14525>

Liang, Q.X., *Governance, CEO Identity, and Quality Provision of Farmer Cooperatives*, Promoter(s): Prof.dr. G.W.J. Hendrikse, EPS-2013-281-ORG, <http://hdl.handle.net/1765/39253>

Liket, K., *Why 'Doing Good' is not Good Enough: Essays on Social Impact Measurement*, Promoter(s): Prof.dr. H.R. Commandeur & Dr. K.E.H. Maas, EPS-2014-307-STR, <http://hdl.handle.net/1765/51130>

Loos, M.J.H.M. van der, *Molecular Genetics and Hormones: New Frontiers in Entrepreneurship Research*, Promoter(s): Prof.dr. A.R. Thurik, Prof.dr. P.J.F. Groenen, & Prof.dr. A. Hofman, EPS-2013-287-S&E, <http://hdl.handle.net/1765/40081>

Lovric, M., *Behavioral Finance and Agent-Based Artificial Markets*, Promoter(s): Prof.dr. J Spronk & Prof.dr.ir. Uzay Kaymak, EPS-2011-229-F&A, <http://hdl.handle.net/1765/22814>

Lu, Y., *Data-Driven Decision Making in Auction Markets*, Promoter(s): Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. W. Ketter, EPS-2014-314-LIS, <http://hdl.handle.net/1765/51543>

Maas, K.E.G., *Corporate Social Performance: From Output Measurement to Impact Measurement*, Promoter(s): Prof.dr. H.R. Commandeur, EPS-2009-182-STR, <http://hdl.handle.net/1765/17627>

Markwat, T.D., *Extreme Dependence in Asset Markets Around the Globe*, Promoter(s): Prof.dr. D.J.C. van Dijk, EPS-2011-227-F&A, <http://hdl.handle.net/1765/22744>

Mees, H., *Changing Fortunes: How China's Boom Caused the Financial Crisis*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2012-266-MKT, <http://hdl.handle.net/1765/34930>

Meuer, J., *Configurations of Inter-firm Relations in Management Innovation: A Study in China's Biopharmaceutical Industry*, Promoter(s): Prof.dr. B. Krug, EPS-2011-228-ORG, <http://hdl.handle.net/1765/22745>

Mihalache, O.R., *Stimulating Firm Innovativeness: Probing the Interrelations between Managerial and Organizational Determinants*, Promoter(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch, & Prof.dr. H.W. Volberda, EPS-2012-260-S&E, <http://hdl.handle.net/1765/32343>

Milea, V., *News Analytics for Financial Decision Support*, Promoter(s): Prof.dr.ir. U. Kaymak, EPS-2013-275-LIS, <http://hdl.handle.net/1765/38673>

Moonen, J.M., *Multi-Agent Systems for Transportation Planning and Coordination*, Promoter(s): Prof.dr. J. van Hillegersberg & Prof.dr. S.L. van de Velde, EPS-2009-177-LIS, <http://hdl.handle.net/1765/16208>

Nederveen Pieterse, A., *Goal Orientation in Teams: The Role of Diversity*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2009-162-ORG, <http://hdl.handle.net/1765/15240>

Nielsen, L.K., *Rolling Stock Rescheduling in Passenger Railways: Applications in short-term planning and in disruption management*, Promoter(s): Prof.dr. L.G. Kroon, EPS-2011-224-LIS, <http://hdl.handle.net/1765/22444>

Niesten, E.M.M.I., *Regulation, Governance and Adaptation: Governance transformations in the Dutch and French liberalizing electricity industries*, Promoter(s): Prof.dr. A. Jolink & Prof.dr. J.P.M. Groenewegen, EPS-2009-170-ORG, <http://hdl.handle.net/1765/16096>

Nijdam, M.H., *Leader Firms: The value of companies for the competitiveness of the Rotterdam seaport cluster*, Promoter(s): Prof.dr. R.J.M. van Tulder, EPS-2010-216-ORG, <http://hdl.handle.net/1765/21405>

Noordegraaf-Eelens, L.H.J., *Contested Communication; A Critical Analysis of Central Bank Speech*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, Prof.dr. J. de Mul, & Prof.dr. D.J.C. van Dijk, EPS-2010-209-MKT, <http://hdl.handle.net/1765/21061>

Nuijten, A.L.P., *Deaf Effect for Risk Warnings: A Causal Examination applied to Information Systems Projects*, Promoter(s): Prof.dr. G.J. van der Pijl, Prof.dr. H.R. Commandeur, & Prof.dr. M. Keil, EPS-2012-263-S&E, <http://hdl.handle.net/1765/34928>

Nuijten, I., *Servant-Leadership: Paradox or Diamond in the Rough? A Multidimensional Measure and Empirical Evidence*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2009-183-ORG, <http://hdl.handle.net/1765/17628>

Oosterhout, M. van, *Business Agility and Information Technology in Service Organizations*, Promoter(s): Prof.dr.ir. H.W.G.M. van Heck, EPS-2010-198-LIS, <http://hdl.handle.net/1765/19805>

Oostrum, J.M. van, *Applying Mathematical Models to Surgical Patient Planning*, Promoter(s): Prof.dr. A.P.M. Wagelmans, EPS-2009-179-LIS, <http://hdl.handle.net/1765/16728>

Osadchiy, S.E., *The Dynamics of Formal Organization: Essays on bureaucracy and formal rules*, Promoter(s): Prof.dr. P.P.M.A.R. Heugens, EPS-2011-231-ORG, <http://hdl.handle.net/1765/23250>

Otgaard, A.H.J., *Industrial Tourism: Where the Public Meets the Private*, Promoter(s): Prof.dr. L. Berg, EPS-2010-219-ORG, <http://hdl.handle.net/1765/21585>

Ozdemir, M.N., *Project-level Governance, Monetary Incentives, and Performance in Strategic R&D Alliances*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2011-235-LIS, <http://hdl.handle.net/1765/23550>

Peers, Y., *Econometric Advances in Diffusion Models*, Promoter(s): Prof.dr. Ph.H.B.F Franses, EPS-2011-251-MKT, <http://hdl.handle.net/1765/30586>

Pinçé, C., *Advances in Inventory Management: Dynamic Models*, Promoter(s): Prof.dr.ir. R. Dekker, EPS-2010-199-LIS, <http://hdl.handle.net/1765/19867>

Porck, J., *No Team is an Island: An Integrative View of Strategic Consensus between Groups*, Promoter(s): Prof.dr. P.J.F. Groenen & Prof.dr. D.L. van Knippenberg, EPS-2013-299-ORG, <http://hdl.handle.net/1765/50141>

Porras Prado, M., *The Long and Short Side of Real Estate, Real Estate Stocks, and Equity*, Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2012-254-F&A, <http://hdl.handle.net/1765/30848>

Poruthiyil, P.V., *Steering Through: How organizations negotiate permanent uncertainty and unresolvable choices*, Promoter(s): Prof.dr. P.M.A.R. Heugens & Prof.dr. S. Magala, EPS-2011-245-ORG, <http://hdl.handle.net/1765/26392>

Potthoff, D., *Railway Crew Rescheduling: Novel approaches and extensions*, Promoter(s): Prof.dr. A.P.M. Wagelmans & Prof.dr. L.G. Kroon, EPS-2010-210-LIS, <http://hdl.handle.net/1765/21084>

Pourakbar, M., *End-of-Life Inventory Decisions of Service Parts*, Promoter(s): Prof.dr.ir. R. Dekker, EPS-2011-249-LIS, <http://hdl.handle.net/1765/30584>

Pronker, E.S., *Innovation Paradox in Vaccine Target Selection*, Promoter(s): Prof.dr. H.J.H.M. Claassen & Prof.dr. H.R. Commandeur, EPS-2013-282-S&E, <http://hdl.handle.net/1765/39654>

Retel Helmrich, M.J., *Green Lot-Sizing*, Promoter(s): Prof.dr. A.P.M. Wagelmans, EPS-2013-291-LIS, <http://hdl.handle.net/1765/41330>

Rijnsbilt, J.A., *CEO Narcissism: Measurement and Impact*, Promoter(s): Prof.dr. A.G.Z. Kemna & Prof.dr. H.R. Commandeur, EPS-2011-238-STR, <http://hdl.handle.net/1765/23554>

Roelofsen, E.M., *The Role of Analyst Conference Calls in Capital Markets*, Promoter(s): Prof.dr. G.M.H. Mertens & Prof.dr. L.G. van der Tas, EPS-2010-190-F&A, <http://hdl.handle.net/1765/18013>

Rosmalen, J. van, *Segmentation and Dimension Reduction: Exploratory and Model-Based Approaches*, Promoter(s): Prof.dr. P.J.F. Groenen, EPS-2009-165-MKT, <http://hdl.handle.net/1765/15536>

Roza-van Vuren, M.W., *The Relationship between Offshoring Strategies and Firm Performance: Impact of innovation, absorptive capacity and firm size*, Promoter(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2011-214-STR, <http://hdl.handle.net/1765/22155>

Rubbaniy, G., *Investment Behaviour of Institutional Investors*, Promoter(s): Prof.dr. W.F.C. Verschoor, EPS-2013-284-F&A, <http://hdl.handle.net/1765/40068>

Rus, D., *The Dark Side of Leadership: Exploring the Psychology of Leader Self-serving Behavior*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2009-178-ORG, <http://hdl.handle.net/1765/16726>

Santos Nogueira, R.J.A. e, *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*, Promoter(s): Prof.dr.ir. U. Kaymak & Prof.dr. J.M.C. Sousa, EPS-2014-310-LIS, <http://hdl.handle.net/1765/51560>

Schellekens, G.A.C., *Language Abstraction in Word of Mouth*, Promoter(s): Prof.dr.ir. A. Smidts, EPS-2010-218-MKT, <http://hdl.handle.net/1765/21580>

Shahzad, K., *Credit Rating Agencies, Financial Regulations and the Capital Markets*, Promoter(s): Prof.dr. G.M.H. Mertens, EPS-2013-283-F&A, <http://hdl.handle.net/1765/39655>

Sotgiu, F., *Not All Promotions are Made Equal: From the Effects of a Price War to Cross-chain Cannibalization*, Promoter(s): Prof.dr. M.G. Dekimpe & Prof.dr.ir. B. Wierenga, EPS-2010-203-MKT, <http://hdl.handle.net/1765/19714>

Sousa, M.J.C. de, *Servant Leadership to the Test: New Perspectives and Insights*, Promoter(s): Prof.dr. D.L. van Knippenberg & Dr. D. van Dierendonck, EPS-2014-313-ORG, <http://hdl.handle.net/1765/51537>

Spliet, R., *Vehicle Routing with Uncertain Demand*, Promoter(s): Prof.dr.ir R. Dekker, EPS-2013-293-LIS, <http://hdl.handle.net/1765/41513>

Srour, F.J., *Dissecting Drayage: An Examination of Structure, Information, and Control in Drayage Operations*, Promoter(s): Prof.dr. S.L. van de Velde, EPS-2010-186-LIS, <http://hdl.handle.net/1765/18231>

Staadtt, J.L., *Leading Public Housing Organisation in a Problematic Situation: A Critical Soft Systems Methodology Approach*, Promoter(s): Prof.dr. S.J. Magala, EPS-2014-308-ORG, <http://hdl.handle.net/1765/50712>

Stallen, M., *Social Context Effects on Decision-Making: A Neurobiological Approach*, Promoter(s): Prof.dr.ir. A. Smidts, EPS-2013-285-MKT, <http://hdl.handle.net/1765/39931>

Sweldens, S.T.L.R., *Evaluative Conditioning 2.0: Direct versus Associative Transfer of Affect to Brands*, Promoter(s): Prof.dr. S.M.J. van Osselaer, EPS-2009-167-MKT, <http://hdl.handle.net/1765/16012>

Tarakci, M., *Behavioral Strategy: Strategic Consensus, Power and Networks*, Promoter(s): Prof.dr. D.L. van Knippenberg & Prof.dr. P.J.F. Groenen, EPS-2013-280-ORG, <http://hdl.handle.net/1765/39130>

Teixeira de Vasconcelos, M., *Agency Costs, Firm Value, and Corporate Investment*, Promoter(s): Prof.dr. P.G.J. Roosenboom, EPS-2012-265-F&A, <http://hdl.handle.net/1765/37265>

Tempelaar, M.P., *Organizing for Ambidexterity: Studies on the pursuit of exploration and exploitation through differentiation, integration, contextual and individual attributes*,

Promoter(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-191-STR, <http://hdl.handle.net/1765/18457>

Tiwari, V., *Transition Process and Performance in IT Outsourcing: Evidence from a Field Study and Laboratory Experiments*, Promoter(s): Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. P.H.M. Vervest, EPS-2010-201-LIS, <http://hdl.handle.net/1765/19868>

Tröster, C., *Nationality Heterogeneity and Interpersonal Relationships at Work*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2011-233-ORG, <http://hdl.handle.net/1765/23298>

Tsekouras, D., *No Pain No Gain: The Beneficial Role of Consumer Effort in Decision-Making*, Promoter(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2012-268-MKT, <http://hdl.handle.net/1765/37542>

Tzioti, S., *Let Me Give You a Piece of Advice: Empirical Papers about Advice Taking in Marketing*, Promoter(s): Prof.dr. S.M.J. van Osselaer & Prof.dr.ir. B. Wierenga, EPS-2010-211-MKT, <http://hdl.handle.net/1765/21149>

Vaccaro, I.G., *Management Innovation: Studies on the Role of Internal Change Agents*, Promoter(s): Prof.dr.ing. F.A.J. van den Bosch, Prof.dr. H.W. Volberda, & Prof.dr. J.J.P. Jansen, EPS-2010-212-STR, <http://hdl.handle.net/1765/21150>

Vagias, D., *Liquidity, Investors and International Capital Markets*, Promoter(s): Prof.dr. M.A. van Dijk, EPS-2013-294-F&A, <http://hdl.handle.net/1765/41511>

Venus, M., *Demystifying Visionary Leadership: In search of the essence of effective vision communication*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2013-289-ORG, <http://hdl.handle.net/1765/40079>

Verheijen, H.J.J., *Vendor-Buyer Coordination in Supply Chains*, Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen, EPS-2010-194-LIS, <http://hdl.handle.net/1765/19594>

Visser, V.A., *Leader Affect and Leadership Effectiveness: How leader affective displays influence follower outcomes*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2013-286-ORG, <http://hdl.handle.net/1765/40076>

Vlam, A.J., *Customer First? The Relationship between Advisors and Consumers of Financial Products*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2011-250-MKT, <http://hdl.handle.net/1765/30585>

Waard, E.J. de, *Engaging Environmental Turbulence: Organizational Determinants for Repetitive Quick and Adequate Responses*, Promoter(s): Prof.dr. H.W. Volberda & Prof.dr. J. Soeters, EPS-2010-189-STR, <http://hdl.handle.net/1765/18012>

Wall, R.S., *NETSCAPE: Cities and Global Corporate Networks*, Promoter(s): Prof.dr. G.A. van der Knaap, EPS-2009-169-ORG, <http://hdl.handle.net/1765/16013>

Waltman, L., *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*, Promoter(s): Prof.dr.ir. R. Dekker & Prof.dr.ir. U. Kaymak, EPS-2011-248-LIS, <http://hdl.handle.net/1765/26564>

Wang, Y., *Information Content of Mutual Fund Portfolio Disclosure*, Promoter(s): Prof.dr. M.J.C.M Verbeek, EPS-2011-242-F&A, <http://hdl.handle.net/1765/26066>

Wang, Y., *Corporate Reputation Management: Reaching Out to Financial Stakeholders*, Promoter(s): Prof.dr. C.B.M. van Riel, EPS-2013-271-ORG, <http://hdl.handle.net/1765/38675>

Weenen, T.C., *On the Origin and Development of the Medical Nutrition Industry*, Promoter(s): Prof.dr. H.R. Commandeur & Prof.dr. H.J.H.M. Claassen, EPS-2014-309-S&E, <http://hdl.handle.net/1765/51134>

Weerdt, N.P. van der, *Organizational Flexibility for Hypercompetitive Markets*, Promoter(s): Prof.dr. H.W. Volberda & Dr. E. Verwaal, EPS-2009-173-STR, <http://hdl.handle.net/1765/16182>

Wolfswinkel, M., *Corporate Governance, Firm Risk and Shareholder Value*, Promoter(s): Prof.dr. A. de Jong, EPS-2013-277-F&A, <http://hdl.handle.net/1765/39127>

Wubben, M.J.J., *Social Functions of Emotions in Social Dilemmas*, Promoter(s): Prof.dr. D. de Cremer & Prof.dr. E. van Dijk, EPS-2010-187-ORG, <http://hdl.handle.net/1765/18228>

Xu, Y., *Empirical Essays on the Stock Returns, Risk Management, and Liquidity Creation of Banks*, Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2010-188-F&A, <http://hdl.handle.net/1765/18125>

Yang, J., *Towards the Restructuring and Co-ordination Mechanisms for the Architecture of Chinese Transport Logistics*, Promoter(s): Prof.dr. H.E. Haralambides, EPS-2009-157-LIS, <http://hdl.handle.net/1765/14527>

Zaerpour, N., *Efficient Management of Compact Storage Systems*, Promoter(s): Prof.dr.ir. M.B.M. de Koster, EPS-2013-276-LIS, <http://hdl.handle.net/1765/38766>

Zhang, D., *Essays in Executive Compensation*, Promoter(s): Prof.dr. I. Dittman, EPS-2012-261-F&A, <http://hdl.handle.net/1765/32344>

Zhang, X., *Scheduling with Time Lags*, Promoter(s): Prof.dr. S.L. van de Velde, EPS-2010-206-LIS, <http://hdl.handle.net/1765/19928>

Zhou, H., *Knowledge, Entrepreneurship and Performance: Evidence from country-level and firm-level studies*, Promoter(s): Prof.dr. A.R. Thurik & Prof.dr. L.M. Uhlaner, EPS-2010-207-ORG, <http://hdl.handle.net/1765/20634>

Zwan, P.W. van der, *The Entrepreneurial Process: An International Analysis of Entry and Exit*, Promoter(s): Prof.dr. A.R. Thurik & Prof.dr. P.J.F. Groenen, EPS-2011-234-ORG, <http://hdl.handle.net/1765/23422>



ERIM PhD Series Research in Management

Erasmus Research Institute of Management - ERIM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
Erasmus University Rotterdam (EUR)
P.O. Box 1738, 3000 DR Rotterdam,
The Netherlands

Tel. +31 10 408 11 82
Fax +31 10 408 96 40
E-mail info@erim.eur.nl
Internet www.erim.eur.nl

ERIM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are the Rotterdam School of Management (RSM), and the Erasmus School of Economics (ESE). ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focused on the management of the firm in its environment, its intra- and interfirrm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced doctoral programme in Research in Management. Within ERIM, over three hundred senior researchers and PhD candidates are active in the different research programmes. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.

AUTOMATED DETECTION OF FINANCIAL EVENTS IN NEWS TEXT

Today's financial markets are inextricably linked with financial events like acquisitions, profit announcements, or product launches. Information extracted from news messages that report on such events could hence be beneficial for financial decision making. The ubiquity of news, however, makes manual analysis impossible, and due to the unstructured nature of text, the (semi-)automatic extraction and application of financial events remains a non-trivial task. Therefore, the studies composing this dissertation investigate 1) how to accurately identify financial events in news text, and 2) how to effectively use such extracted events in financial applications.

Based on a detailed evaluation of current event extraction systems, this thesis presents a competitive, knowledge-driven, semi-automatic system for financial event extraction from text. A novel pattern language, which makes clever use of the system's underlying knowledge base, allows for the definition of simple, yet expressive event extraction rules that can be applied to natural language texts. The system's knowledge-driven internals remain synchronized with the latest market developments through the accompanying event-triggered update language for knowledge bases, enabling the definition of update rules.

Additional research covered by this dissertation investigates the practical applicability of extracted events. In automated stock trading experiments, the best performing trading rules do not only make use of traditional numerical signals, but also employ news-based event signals. Moreover, when cleaning stock data from disruptions caused by financial events, financial risk analyses yield more accurate results. These results suggest that events detected in news can be used advantageously as supplementary parameters in financial applications.