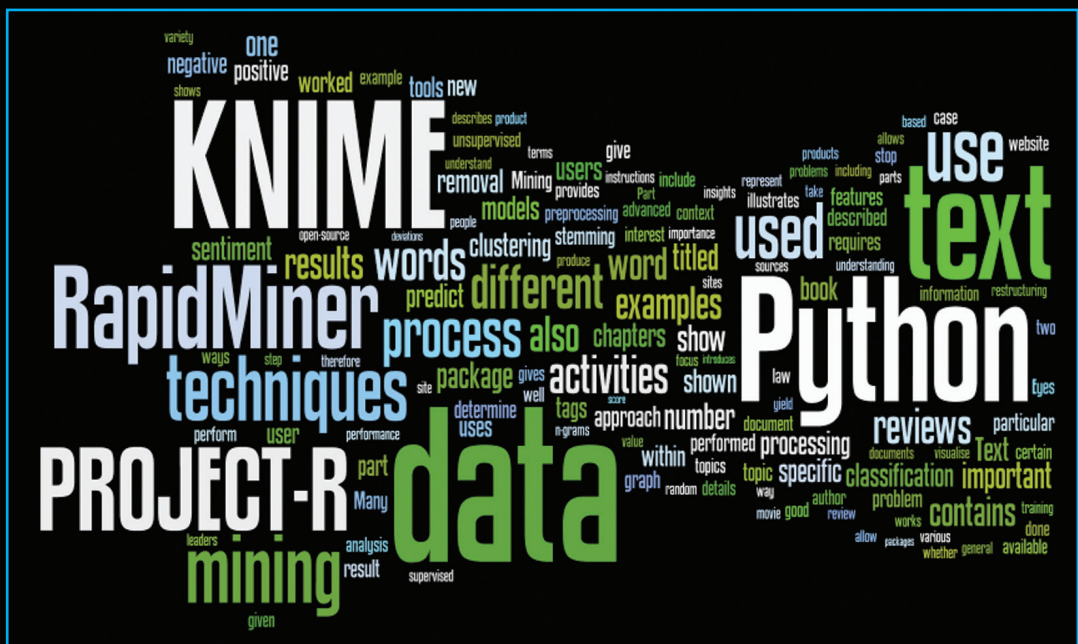


# TEXT MINING AND VISUALIZATION

# Case Studies Using Open-Source Tools



**Edited by**  
**Markus Hofmann**  
**Andrew Chisholm**



CRC Press

Taylor &amp; Francis Group

A CHAPMAN &amp; HALL BOOK

# **TEXT MINING AND VISUALIZATION**

Case Studies Using  
Open-Source Tools

**Chapman & Hall/CRC**  
**Data Mining and Knowledge Discovery Series**

**SERIES EDITOR**

**Vipin Kumar**

University of Minnesota  
Department of Computer Science and Engineering  
Minneapolis, Minnesota, U.S.A.

**AIMS AND SCOPE**

This series aims to capture new developments and applications in data mining and knowledge discovery, while summarizing the computational tools and techniques useful in data analysis. This series encourages the integration of mathematical, statistical, and computational methods and techniques through the publication of a broad range of textbooks, reference works, and handbooks. The inclusion of concrete examples and applications is highly encouraged. The scope of the series includes, but is not limited to, titles in the areas of data mining and knowledge discovery methods and applications, modeling, algorithms, theory and foundations, data and knowledge visualization, data mining systems and tools, and privacy and security issues.

**PUBLISHED TITLES**

**ACCELERATING DISCOVERY : MINING UNSTRUCTURED INFORMATION FOR  
HYPOTHESIS GENERATION**

Scott Spangler

**ADVANCES IN MACHINE LEARNING AND DATA MINING FOR ASTRONOMY**

Michael J. Way, Jeffrey D. Scargle, Kamal M. Ali, and Ashok N. Srivastava

**BIOLOGICAL DATA MINING**

Jake Y. Chen and Stefano Lonardi

**COMPUTATIONAL BUSINESS ANALYTICS**

Subrata Das

**COMPUTATIONAL INTELLIGENT DATA ANALYSIS FOR SUSTAINABLE  
DEVELOPMENT**

Ting Yu, Nitesh V. Chawla, and Simeon Simoff

**COMPUTATIONAL METHODS OF FEATURE SELECTION**

Huan Liu and Hiroshi Motoda

**CONSTRAINED CLUSTERING: ADVANCES IN ALGORITHMS, THEORY,  
AND APPLICATIONS**

Sugato Basu, Ian Davidson, and Kiri L. Wagstaff

**CONTRAST DATA MINING: CONCEPTS, ALGORITHMS, AND APPLICATIONS**

Guozhu Dong and James Bailey

**DATA CLASSIFICATION: ALGORITHMS AND APPLICATIONS**

Charu C. Aggarawal

**DATA CLUSTERING: ALGORITHMS AND APPLICATIONS**

Charu C. Aggarawal and Chandan K. Reddy

**DATA CLUSTERING IN C++: AN OBJECT-ORIENTED APPROACH**

Guojun Gan

**DATA MINING FOR DESIGN AND MARKETING**

Yukio Ohsawa and Katsutoshi Yada

**DATA MINING WITH R: LEARNING WITH CASE STUDIES**

Luís Torgo

**EVENT MINING: ALGORITHMS AND APPLICATIONS**

Tao Li

**FOUNDATIONS OF PREDICTIVE ANALYTICS**

James Wu and Stephen Coggeshall

**GEOGRAPHIC DATA MINING AND KNOWLEDGE DISCOVERY,  
SECOND EDITION**

Harvey J. Miller and Jiawei Han

**GRAPH-BASED SOCIAL MEDIA ANALYSIS**

Ioannis Pitas

**HANDBOOK OF EDUCATIONAL DATA MINING**

Cristóbal Romero, Sebastian Ventura, Mykola Pechenizkiy, and Ryan S.J.d. Baker

**HEALTHCARE DATA ANALYTICS**

Chandan K. Reddy and Charu C. Aggarwal

**INFORMATION DISCOVERY ON ELECTRONIC HEALTH RECORDS**

Vagelis Hristidis

**INTELLIGENT TECHNOLOGIES FOR WEB APPLICATIONS**

Priti Srinivas Sajja and Rajendra Akerkar

**INTRODUCTION TO PRIVACY-PRESERVING DATA PUBLISHING: CONCEPTS  
AND TECHNIQUES**

Benjamin C. M. Fung, Ke Wang, Ada Wai-Chee Fu, and Philip S. Yu

**KNOWLEDGE DISCOVERY FOR COUNTERTERRORISM AND  
LAW ENFORCEMENT**

David Skillicorn

**KNOWLEDGE DISCOVERY FROM DATA STREAMS**

João Gama

**MACHINE LEARNING AND KNOWLEDGE DISCOVERY FOR  
ENGINEERING SYSTEMS HEALTH MANAGEMENT**

Ashok N. Srivastava and Jiawei Han

**MINING SOFTWARE SPECIFICATIONS: METHODOLOGIES AND APPLICATIONS**

David Lo, Siau-Cheng Khoo, Jiawei Han, and Chao Liu

**MULTIMEDIA DATA MINING: A SYSTEMATIC INTRODUCTION TO  
CONCEPTS AND THEORY**

Zhongfei Zhang and Ruofei Zhang

**MUSIC DATA MINING**

Tao Li, Mitsunori Ogihara, and George Tzanetakis

**NEXT GENERATION OF DATA MINING**

Hillol Kargupta, Jiawei Han, Philip S. Yu, Rajeev Motwani, and Vipin Kumar

**RAPIDMINER: DATA MINING USE CASES AND BUSINESS ANALYTICS  
APPLICATIONS**

Markus Hofmann and Ralf Klinkenberg

**RELATIONAL DATA CLUSTERING: MODELS, ALGORITHMS,  
AND APPLICATIONS**

Bo Long, Zhongfei Zhang, and Philip S. Yu

**SERVICE-ORIENTED DISTRIBUTED KNOWLEDGE DISCOVERY**

Domenico Talia and Paolo Trunfio

**SPECTRAL FEATURE SELECTION FOR DATA MINING**

Zheng Alan Zhao and Huan Liu

**STATISTICAL DATA MINING USING SAS APPLICATIONS, SECOND EDITION**

George Fernandez

**SUPPORT VECTOR MACHINES: OPTIMIZATION BASED THEORY,  
ALGORITHMS, AND EXTENSIONS**

Naiyang Deng, Yingjie Tian, and Chunhua Zhang

**TEMPORAL DATA MINING**

Theophano Mitsa

**TEXT MINING: CLASSIFICATION, CLUSTERING, AND APPLICATIONS**

Ashok N. Srivastava and Mehran Sahami

**TEXT MINING AND VISUALIZATION: CASE STUDIES USING OPEN-SOURCE  
TOOLS**

Markus Hofmann and Andrew Chisholm

**THE TOP TEN ALGORITHMS IN DATA MINING**

Xindong Wu and Vipin Kumar

**UNDERSTANDING COMPLEX DATASETS: DATA MINING WITH MATRIX  
DECOMPOSITIONS**

David Skillicorn

# TEXT MINING AND VISUALIZATION

Case Studies Using  
Open-Source Tools

Edited by  
**Markus Hofmann**  
**Andrew Chisholm**



**CRC Press**

Taylor & Francis Group  
Boca Raton London New York

---

CRC Press is an imprint of the  
Taylor & Francis Group, an **informa** business  
A CHAPMAN & HALL BOOK

CRC Press  
Taylor & Francis Group  
6000 Broken Sound Parkway NW, Suite 300  
Boca Raton, FL 33487-2742

© 2016 by Taylor & Francis Group, LLC  
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works  
Version Date: 20151105

International Standard Book Number-13: 978-1-4822-3758-0 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access [www.copyright.com](http://www.copyright.com) (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

**Trademark Notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

**Visit the Taylor & Francis Web site at**  
**<http://www.taylorandfrancis.com>**

**and the CRC Press Web site at**  
**<http://www.crcpress.com>**

---

## ***Dedication - Widmung***

Für meine Großeltern, Luise and Matthias Hofmann - Danke für ALLES!

Euer Enkel, Markus

To Jennie

Andrew



This page intentionally left blank

---

# Contents

<b>I</b>	<b>RapidMiner</b>	<b>1</b>
<b>1</b>	<b>RapidMiner for Text Analytic Fundamentals</b>	<b>3</b>
	<i>John Ryan</i>	
1.1	Introduction . . . . .	3
1.2	Objectives . . . . .	4
1.2.1	Education Objective . . . . .	4
1.2.2	Text Analysis Task Objective . . . . .	4
1.3	Tools Used . . . . .	4
1.4	First Procedure: Building the Corpus . . . . .	5
1.4.1	Overview . . . . .	5
1.4.2	Data Source . . . . .	5
1.4.3	Creating Your Repository . . . . .	5
1.4.3.1	Download Information from the Internet . . . . .	5
1.5	Second Procedure: Build a Token Repository . . . . .	11
1.5.1	Overview . . . . .	11
1.5.2	Retrieving and Extracting Text Information . . . . .	11
1.5.3	Summary . . . . .	17
1.6	Third Procedure: Analyzing the Corpus . . . . .	20
1.6.1	Overview . . . . .	20
1.6.2	Mining Your Repository — Frequency of Words . . . . .	20
1.6.3	Mining Your Repository — Frequency of N-Grams . . . . .	25
1.6.4	Summary . . . . .	27
1.7	Fourth Procedure: Visualization . . . . .	27
1.7.1	Overview . . . . .	27
1.7.2	Generating word clouds . . . . .	27
1.7.2.1	Visualize Your Data . . . . .	27
1.7.3	Summary . . . . .	35
1.8	Conclusion . . . . .	35
<b>2</b>	<b>Empirical Zipf-Mandelbrot Variation for Sequential Windows within Documents</b>	<b>37</b>
	<i>Andrew Chisholm</i>	
2.1	Introduction . . . . .	37
2.2	Structure of This Chapter . . . . .	38
2.3	Rank–Frequency Distributions . . . . .	39
2.3.1	Heaps’ Law . . . . .	39
2.3.2	Zipf’s Law . . . . .	39
2.3.3	Zipf-Mandelbrot . . . . .	40
2.4	Sampling . . . . .	41
2.5	RapidMiner . . . . .	42
2.5.1	Creating Rank–Frequency Distributions . . . . .	43

2.5.1.1	Read Document and Create Sequential Windows . . . . .	43
2.5.1.2	Create Rank–Frequency Distribution for Whole Document and Most Common Word List . . . . .	46
2.5.1.3	Calculate Rank–Frequency Distributions for Most Common Words within Sequential Windows . . . . .	47
2.5.1.4	Combine Whole Document with Sequential Windows . . .	47
2.5.2	Fitting Zipf-Mandelbrot to a Distribution: Iterate . . . . .	47
2.5.3	Fitting Zipf-Mandelbrot to a Distribution: Fitting . . . . .	48
2.6	Results . . . . .	52
2.6.1	Data . . . . .	52
2.6.2	Starting Values for Parameters . . . . .	52
2.6.3	Variation of Zipf-Mandelbrot Parameters by Distance through Documents . . . . .	53
2.7	Discussion . . . . .	58
2.8	Summary . . . . .	59
II	<b>KNIME</b>	<b>61</b>
3	<b>Introduction to the KNIME Text Processing Extension</b>	<b>63</b>
	<i>Kilian Thiel</i>	
3.1	Introduction . . . . .	63
3.1.1	Installation . . . . .	64
3.2	Philosophy . . . . .	64
3.2.1	Reading Textual Data . . . . .	66
3.2.2	Enrichment and Tagging . . . . .	66
3.2.2.1	Unmodifiability . . . . .	68
3.2.2.2	Concurrency . . . . .	68
3.2.2.3	Tagger Conflicts . . . . .	68
3.2.3	Preprocessing . . . . .	70
3.2.3.1	Preprocessing Dialog . . . . .	70
3.2.4	Frequencies . . . . .	72
3.2.5	Transformation . . . . .	72
3.3	Data Types . . . . .	73
3.3.1	Document Cell . . . . .	73
3.3.2	Term Cell . . . . .	74
3.4	Data Table Structures . . . . .	74
3.5	Example Application: Sentiment Classification . . . . .	75
3.5.1	Reading Textual Data . . . . .	76
3.5.2	Preprocessing . . . . .	76
3.5.3	Transformation . . . . .	77
3.5.4	Classification . . . . .	77
3.6	Summary . . . . .	79
4	<b>Social Media Analysis — Text Mining Meets Network Mining</b>	<b>81</b>
	<i>Kilian Thiel, Tobias Kötter, Rosaria Silipo, and Phil Winters</i>	
4.1	Introduction . . . . .	81
4.2	The Slashdot Data Set . . . . .	82
4.3	Text Mining the Slashdot Data . . . . .	82
4.4	Network Mining the Slashdot Data . . . . .	87
4.5	Combining Text and Network Mining . . . . .	88
4.6	Summary . . . . .	91

<b>III</b>	<b>Python</b>	<b>93</b>
<b>5</b>	<b>Mining Unstructured User Reviews with Python</b>	<b>95</b>
	<i>Brian Carter</i>	
5.1	Introduction . . . . .	95
5.1.1	Workflow of the Chapter . . . . .	97
5.1.2	Scope of Instructions . . . . .	97
5.1.3	Technical Stack Setup . . . . .	99
5.1.3.1	Python . . . . .	99
5.1.3.2	MongoDB . . . . .	100
5.1.4	Python for Text Mining . . . . .	100
5.1.4.1	Sparse Matrices . . . . .	100
5.1.4.2	Text Representation in Python . . . . .	103
5.1.4.3	Character Encoding in Python . . . . .	105
5.2	Web Scraping . . . . .	106
5.2.1	Regular Expression in Python . . . . .	108
5.2.2	PillReports.net Web Scrape . . . . .	108
5.3	Data Cleansing . . . . .	111
5.4	Data Visualization and Exploration . . . . .	114
5.4.1	Python:Matplotlib . . . . .	115
5.4.2	Pillreports.net Data Exploration . . . . .	115
5.5	Classification . . . . .	119
5.5.1	Classification with sklearn . . . . .	120
5.5.2	Classification Results . . . . .	122
5.6	Clustering & PCA . . . . .	124
5.6.1	Word Cloud . . . . .	128
5.7	Summary . . . . .	130
<b>6</b>	<b>Sentiment Classification and Visualization of Product Review Data</b>	<b>133</b>
	<i>Alexander Piazza and Pavlina Davcheva</i>	
6.1	Introduction . . . . .	134
6.2	Process . . . . .	136
6.2.1	Step 1: Preparation and Loading of Data Set . . . . .	137
6.2.2	Step 2: Preprocessing and Feature Extraction . . . . .	138
6.2.3	Step 3: Sentiment Classification . . . . .	142
6.2.4	Step 4: Evaluation of the Trained Classifier . . . . .	143
6.2.5	Step 5: Visualization of Review Data . . . . .	145
6.3	Summary . . . . .	151
<b>7</b>	<b>Mining Search Logs for Usage Patterns</b>	<b>153</b>
	<i>Tony Russell-Rose and Paul Clough</i>	
7.1	Introduction . . . . .	153
7.2	Getting Started . . . . .	154
7.3	Using Clustering to Find Patterns . . . . .	158
7.3.1	Step 1: Prepare the Data . . . . .	159
7.3.2	Step 2: Cluster Using Weka . . . . .	159
7.3.3	Step 3: Visualise the Output . . . . .	161
7.4	Replication and Validation . . . . .	164
7.5	Summary . . . . .	168

<b>8</b>	<b>Temporally Aware Online News Mining and Visualization with Python</b>	<b>173</b>
	<i>Kyle Goslin</i>	
8.1	Introduction . . . . .	174
8.1.1	Section Outline . . . . .	174
8.1.2	Conventions . . . . .	174
8.2	What You Need . . . . .	175
8.2.1	Local HTTP Host . . . . .	175
8.2.2	Python Interpreter . . . . .	175
8.2.3	PIP Installer . . . . .	176
8.2.4	lxml and Scrapy . . . . .	177
8.2.5	SigmaJS Visualization . . . . .	177
8.2.6	News Source . . . . .	178
8.2.7	Python Code Text Editor . . . . .	178
8.3	Crawling and Scraping — What Is the Difference? . . . . .	179
8.4	What Is Time? . . . . .	179
8.5	Analyzing Input Sources . . . . .	180
8.5.1	Parameters for Webpages . . . . .	182
8.5.2	Processing Sections of Webpages vs. Processing Full Webpages . . . . .	182
8.6	Scraping the Web . . . . .	183
8.6.1	Pseudocode for the Crawler . . . . .	184
8.6.2	Creating a Basic Scraping Tool . . . . .	184
8.6.3	Running a Scraping Tool . . . . .	184
8.6.4	News Story Object . . . . .	185
8.6.5	Custom Parsing Function . . . . .	185
8.6.6	Processing an Individual News Story . . . . .	186
8.6.7	Python Imports and Global Variables . . . . .	188
8.7	Generating a Visual Representation . . . . .	188
8.7.1	Generating JSON Data . . . . .	189
8.7.2	Writing a JSON File . . . . .	191
8.8	Viewing the Visualization . . . . .	193
8.9	Additional Concerns . . . . .	194
8.9.1	Authentication . . . . .	196
8.9.2	Errors . . . . .	196
8.10	Summary . . . . .	197
<b>9</b>	<b>Text Classification Using Python</b>	<b>199</b>
	<i>David Colton</i>	
9.1	Introduction . . . . .	199
9.1.1	Python . . . . .	200
9.1.2	The Natural Language Toolkit . . . . .	200
9.1.3	scikit-learn . . . . .	200
9.1.4	Verifying Your Environment . . . . .	201
9.1.5	Movie Review Data Set . . . . .	201
9.1.6	Precision, Recall, and Accuracy . . . . .	202
9.1.7	G-Performance . . . . .	203
9.2	Modelling with the Natural Language Toolkit . . . . .	204
9.2.1	Movie Review Corpus Data Review . . . . .	204
9.2.2	Developing a NLTK Naïve Bayes Classifier . . . . .	205
9.2.3	N-Grams . . . . .	208
9.2.4	Stop Words . . . . .	209

9.2.5	Other Things to Consider . . . . .	210
9.3	Modelling with scikit-learn . . . . .	212
9.3.1	Developing a Naïve Bayes scikit-learn Classifier . . . . .	213
9.3.2	Developing a Support Vector Machine scikit-learn Classifier . . . . .	218
9.4	Conclusions . . . . .	219

## IV R 221

### 10 Sentiment Analysis of Stock Market Behavior from Twitter Using the R Tool 223

*Nuno Oliveira, Paulo Cortez, and Nelson Areal*

10.1	Introduction . . . . .	224
10.2	Methods . . . . .	225
10.3	Data Collection . . . . .	226
10.4	Data Preprocessing . . . . .	227
10.5	Sentiment Analysis . . . . .	230
10.6	Evaluation . . . . .	234
10.7	Inclusion of Emoticons and Hashtags Features . . . . .	236
10.8	Conclusion . . . . .	238

### 11 Topic Modeling 241

*Patrick Buckley*

11.1	Introduction . . . . .	241
11.1.1	Latent Dirichlet Allocation (LDA) . . . . .	242
11.2	Aims of This Chapter . . . . .	243
11.2.1	The Data Set . . . . .	243
11.2.2	R . . . . .	243
11.3	Loading the Corpus . . . . .	244
11.3.1	Import from Local Directory . . . . .	244
11.3.2	Database Connection (RODBC) . . . . .	245
11.4	Preprocessing the Corpus . . . . .	246
11.5	Document Term Matrix (DTM) . . . . .	247
11.6	Creating LDA Topic Models . . . . .	249
11.6.1	Topicmodels Package . . . . .	250
11.6.2	Determining the Optimum Number of Topics . . . . .	251
11.6.3	LDA Models with Gibbs Sampling . . . . .	255
11.6.4	Applying New Data to the Model . . . . .	259
11.6.5	Variational Expectations Maximization (VEM) Inference Technique . . . . .	259
11.6.6	Comparison of Inference Techniques . . . . .	261
11.6.7	Removing Names and Creating an LDA Gibbs Model . . . . .	263
11.7	Summary . . . . .	263

### 12 Empirical Analysis of the Stack Overflow Tags Network 265

*Christos Iraklis Tsatsoulis*

12.1	Introduction . . . . .	265
12.2	Data Acquisition and Summary Statistics . . . . .	267
12.3	Full Graph — Construction and Limited Analysis . . . . .	268
12.4	Reduced Graph — Construction and Macroscopic Analysis . . . . .	272
12.5	Node Importance: Centrality Measures . . . . .	273
12.6	Community Detection . . . . .	277
12.7	Visualization . . . . .	282

12.8 Discussion . . . . . 285

12.9 Appendix: Data Acquisition & Parsing . . . . . 290

**Index** **295**

---

## Foreword

Data analysis has received a lot of attention in recent years and the newly coined data scientist is on everybody's radar. However, in addition to the inherent crop of new buzz words, two fundamental things have changed. Data analysis now relies on more complex and heterogeneous data sources; users are no longer content with analyzing a few numbers. They want to integrate data from different sources, scrutinizing data of diverse types. Almost more importantly, tool providers and users have realized that no single proprietary software vendor can provide the wealth of tools required for the job. This has sparked a huge increase in open-source software used for professional data analysis.

The timing of this book could not be better. It focuses on text mining, text being one of the data sources still to be truly harvested, and on open-source tools for the analysis and visualization of textual data. It explores the top-two representatives of two very different types of tools: programming languages and visual workflow editing environments. R and Python are now in widespread use and allow experts to program highly versatile code for sophisticated analytical tasks. At the other end of the spectrum are visual workflow tools that enable even nonexperts to use predefined templates (or blueprints) and modify analyses. Using a visual workflow has the added benefit that intuitive documentation and guidance through the process is created implicitly. RapidMiner (version 5.3, which is still open source) and KNIME are examples of these types of tools. It is worth noting that especially the latter stands on the shoulders of giants: KNIME integrates not only R and Python but also various libraries. (Stanford's NLP package and the Apache openNLP project, among others, are examined more closely in the book.) These enable the use of state-of-the-art methods via an easy-to-use graphical workflow editor.

In a way, the four parts of this book could therefore be read front to back. The reader starts with a visual workbench, assembling complex analytical workflows. But when a certain method is missing, the user can draw on the preferred analytical scripting language to access bleeding-edge technology that has not yet been exposed natively as a visual component. The reverse order also works. Expert coders can continue to work the way they like to work by quickly writing efficient code, and at the same time they can wrap their code into visual components and make that wisdom accessible to nonexperts as well!

Markus and Andrew have done an outstanding job bringing together this volume of both introductory and advanced material about text mining using modern open source technology in a highly accessible way.

**Prof. Dr. Michael Berthold** (*University Konstanz, Germany*)



This page intentionally left blank

---

# Preface

When people communicate, they do it in lots of ways. They write books and articles, create blogs and webpages, interact by sending messages in many different ways, and of course they speak to one another. When this happens electronically, these text data become very accessible and represent a significant and increasing resource that has tremendous potential value to a wide range of organisations. This is because text data represent what people are thinking or feeling and with whom they are interacting, and thus can be used to predict what people will do, how they are feeling about a particular product or issue, and also who else in their social group could be similar. The process of extracting value from text data, known as *text mining*, is the subject of this book.

There are challenges, of course. In recent years, there has been an undeniable explosion of text data being produced from a multitude of sources in large volumes and at great speed. This is within the context of the general huge increases in all forms of data. This volume and variety require new techniques to be applied to the text data to deal with them effectively. It is also true that text data by their nature tend to be *unstructured*, which requires specific techniques to be adopted to clean and restructure them. Interactions between people leads to the formation of networks, and to understand and exploit these requires an understanding of some potentially complex techniques.

It remains true that organisations wishing to exploit text data need new ways of working to stay ahead and to take advantage of what is available. These include general knowledge of the latest and most powerful tools, understanding the data mining process, understanding specific text mining activities, and simply getting an overview of what possibilities there are.

This book provides an introduction to text mining using some of the most popular and powerful open-source tools, *KNIME*, *RapidMiner*, *Weka*, *R*, and *Python*. In addition, the *Many Eyes* website is used to help visualise results. The chapters show text data being gathered and processed from a wide variety of sources, including books, server-access logs, websites, social media sites, and message boards. Each chapter within the book is presented as an example use-case that the reader can follow as part of a step-by-step reproducible example. In the real world, no two problems are the same, and it would be impossible to produce a use case example for every one. However, the techniques, once learned, can easily be applied to other problems and extended. All the examples are downloadable from the website that accompanies this book and the use of open-source tools ensures that they are readily accessible. The book's website is

<http://www.text-mining-book.com>

Text mining is a subcategory within data mining as a whole, and therefore the chapters illustrate a number of data mining techniques including *supervised learning* using classifiers such as *naïve Bayes* and *support vector machines*; *cross-validation* to estimate model per-

formance using a variety of performance measures; and *unsupervised clustering* to partition data into clusters.

Data mining requires significant preprocessing activities such as cleaning, restructuring, and handling missing values. Text mining also requires these activities particularly when text data is extracted from webpages. Text mining also introduces new preprocessing techniques such as tokenizing, stemming, and generation of n-grams. These techniques are amply illustrated in many of the chapters. In addition some novel techniques for applying network methods to text data gathered in the context of message websites are shown.

---

## What Is the Structure of This Book, and Which Chapters Should I Read?

The book consists of four main parts corresponding to the main tools used: RapidMiner, KNIME, Python, and R.

Part 1 about RapidMiner usage contains two chapters. Chapter 1 is titled “RapidMiner for Text Analytic Fundamentals” and is a practical introduction to the use of various open-source tools to perform the basic but important preprocessing steps that are usually necessary when performing any type of text mining exercise. RapidMiner is given particular focus, but the MySQL database and Many Eyes visualisation website are also used. The specific text corpus that is used consists of the inaugural speeches made by US presidents, and the objective of the chapter is to preprocess and import these sufficiently to give visibility to some of the features within them. The speeches themselves are available on the Internet, and the chapter illustrates how to use RapidMiner to access their locations to download the content as well as to parse it so that only the text is used. The chapter illustrates storing the speeches in a database and goes on to show how RapidMiner can be used to perform tasks like tokenising to eliminate punctuation, numbers, and white space as part of building a word vector. Stop word removal using both standard English and a custom dictionary is shown. Creation of word n-grams is also shown as well as techniques for filtering them. The final part of the chapter shows how the Many Eyes online service can take the output from the process to visualise it using a word cloud. At all stages, readers are encouraged to recreate and modify the processes for themselves.

Chapter 2 is more advanced and is titled “Empirical Zipf-Mandelbrot Variation for Sequential Windows within Documents”. It relates to the important area of authorship attribution within text mining. This technique is used to determine the author of a piece of text or sometimes who the author is not. Many attribution techniques exist, and some are based to a certain extent on departures from Zipf’s law. This law states that the rank and frequency of common words when multiplied together yield a constant. Clearly this is a simplification, and the deviations from this for a particular author may reveal a style representative of the author. Modifications to Zipf’s law have been proposed, one of which is the Zipf-Mandelbrot law. The deviations from this law may reveal similarities for works produced by the same author. This chapter uses an advanced RapidMiner process to fit, using a genetic algorithm approach, works by different authors to Zipf-Mandelbrot models and determines the deviations to visualize what similarities there are between authors.

Additionally, an author's work is randomised to produce a random sampling to determine how different the actual works are from a random book to show whether the order of words in a book contributes to an author's style. The results are visualised using R and show some evidence that different authors have similarities of style that is not random.

Part 2 of the book describes the use of the Konstanz Information Miner (KNIME) and again contains two chapters. Chapter 3 introduces the text processing capabilities of KNIME and is titled "Introduction to the KNIME Text Processing Extension". KNIME is a popular open-source platform that uses a visual paradigm to allow processes to be rapidly assembled and executed to allow all data processing, analysis, and mining problems to be addressed. The platform has a plug-in architecture that allows extensions to be installed, and one such is the text processing feature. This chapter describes the installation and use of this extension as part of a text mining process to predict sentiment of movie reviews. The aim of the chapter is to give a good introduction to the use of KNIME in the context of this overall classification process, and readers can use the ideas and techniques for themselves. The chapter gives more background details about the important preprocessing activities that are typically undertaken when dealing with text. These include entity recognition such as the identification of names or other domain-specific items, and tagging parts of speech to identify nouns, verbs, and so on. An important point that is especially relevant as data volumes increase is the possibility to perform processing activities in parallel to take advantage of available processing power, and to reduce the total time to process. Common preprocessing activities such as stemming, number removal, punctuation, handling small and stop words that are described in other chapters with other tools can also be performed with KNIME. The concepts of documents and the bag of words representation are described and the different types of word or document vectors that can be produced are explained. These include term frequencies but can use inverse document frequencies if the problem at hand requires it. Having described the background, the chapter then uses the techniques to build a classifier to predict positive or negative movie reviews based on available training data. This shows use of other parts of KNIME to build a classifier on training data, to apply it to test data, and to observe the accuracy of the prediction.

Chapter 4 is titled "Social Media Analysis — Text Mining Meets Network Mining" and presents a more advanced use of KNIME with a novel way to combine sentiment of users with how they are perceived as influencers in the Slashdot online forum. The approach is motivated by the marketing needs that companies have to identify users with certain traits and find ways to influence them or address the root causes of their views. With the ever increasing volume and types of online data, this is a challenge in its own right, which makes finding something actionable in these fast-moving data sources difficult. The chapter has two parts that combine to produce the result. First, a process is described that gathers user reviews from the Slashdot forum to yield an attitude score for each user. This score is the difference between positive and negative words, which is derived from a lexicon, the MPQA subjectivity lexicon in this case, although others could be substituted as the domain problem dictates. As part of an exploratory confirmation, a tag cloud of words used by an individual user is also drawn where negative and positive words are rendered in different colours. The second part of the chapter uses network analysis to find users who are termed leaders and those who are followers. A leader is one whose published articles gain more comments from others, whereas a follower is one who tends to comment more. This is done in KNIME by using the HITS algorithm often used to rate webpages. In this case, users take the place of websites, and authorities become equivalent to leaders and hubs followers. The two different views are then combined to determine the characteristics of leaders compared with followers from an attitude perspective. The result is that leaders tend to score more

highly on attitude; that is, they are more positive. This contradicts the normal marketing wisdom that negative sentiment tends to be more important.

Part 3 contains five chapters that focus on a wide variety of use cases. Chapter 5 is titled “Mining Unstructured User Reviews with Python” and gives a detailed worked example of mining another social media site where reviews of drugs are posted by users. The site, [pillreports.com](http://pillreports.com), does not condone the use of drugs but provides a service to alert users to potentially life-threatening problems found by real users. The reviews are generally short text entries and are often tagged with a good or bad review. This allows for classification models to be built to try and predict the review in cases where none is provided. In addition, an exploratory clustering is performed on the review data to determine if there are features of interest. The chapter is intended to be illustrative of the techniques and tools that can be used and starts with the process of gathering the data from the Pill Reports website. Python is used to navigate and select the relevant text for storage in a MongoDB datastore. It is the nature of Web scraping that it is very specific to a site and can be fairly involved; the techniques shown will therefore be applicable to other sites. The cleaning and restructuring activities that are required are illustrated with worked examples using Python, including reformatting dates, removing white space, stripping out HTML tags, renaming columns, and generation of n-grams. As a precursor to the classification task to aid understanding of the data, certain visualisation and exploration activities are described. The Python Matplotlib package is used to visualise results, and examples are given. The importance of restructuring the data using grouping and aggregation techniques to get the best out of the visualisations is stressed with details to help. Moving on to the classification step, simple classifiers are built to predict the positive or negative reviews. The initial results are improved through feature selection, and the top terms that predict the class are shown. This is very typical of the sorts of activities that are undertaken during text mining and classification in general, and the techniques will therefore be reusable in other contexts. The final step is to cluster the reviews to determine if there is some unseen structure of interest. This is done using a combination of k-means clustering and principal component analysis. Visualising the results allows a user to see if there are patterns of interest.

Chapter 6 titled “Sentiment Classification and Visualization of Product Review Data” is about using text data gathered from website consumer reviews of products to build a model that can predict sentiment. The difficult problem of obtaining training data is addressed by using the star ratings generally given to products as a proxy for whether the product is good or bad. The motivation for this is to allow companies to assess how well particular products are being received in the market. The chapter aims to give worked examples with a focus on illustrating the end-to-end process rather than the specific accuracy of the techniques tried. Having said that, however, accuracies in excess of 80 percent are achieved for certain product categories. The chapter makes extensive use of Python with the NumPy, NLTK, and Scipy packages, and includes detailed worked examples. As with all data mining activities, extensive data preparation is required, and the chapter illustrates the important steps required. These include, importing correctly from webpages to ensure only valid text is used, tokenizing to find words used in unigrams or bigrams, removal of stop words and punctuation, and stemming and changing emoticons to text form. The chapter then illustrates production of classification models to determine if the extracted features can predict the sentiment expressed from the star rating. The classification models produce interesting results, but to go further and understand what contributes to the positive and negative sentiment, the chapter also gives examples using the open-source Many Eyes tool to show different visualisations and perspectives on the data. This would be valuable for product vendors wanting to gain insight into the reviews of their products.

Chapter 7 “Mining Search Logs for Usage Patterns” is about mining transaction logs containing information about the details of searches users have performed and shows how unsupervised clustering can be performed to identify different types of user. The insights could help to drive services and applications of the future. Given the assumption that what a user searches for is a good indication of his or her intent, the chapter draws together some of the important contributions in this area and proceeds with an example process to show this working in a real context. The specific data that are processed are search transaction data from AOL, and the starting point is to extract a small number of features of interest. These are suggested from similar works, and the first step is to process the logs to represent the data with these features. This is done using Python, and examples are given. The open-source tool Weka is then used to perform an unsupervised clustering using expectation maximization to yield a candidate “best” clustering. As with all clustering techniques and validity measures, the presented answer is not necessarily the best in terms of fit to the problem domain. However, there is value because it allows the user to focus and use intelligent reasoning to understand what the result is showing and what additional steps would be needed to improve the model. This is done in the chapter where results are considered, alternative features are considered and different processing is performed with the end result that a more convincing case is made for the final answer. On the way, the importance of visualising the results, repeating to check that the results are repeatable, and being sceptical are underlined. The particular end result is of interest, but more importantly, it is the process that has been followed that gives the result more power. Generally speaking, this chapter supports the view that a process approach that is iterative in nature is the way to achieve strong results.

Chapter 8, “Temporally Aware Online News Mining and Visualization with Python”, discusses how some sources of text data such as newsfeeds or reviews can have more significance if the information is more recent. With this in mind, this chapter introduces time into text mining. The chapter contains very detailed instructions on how to crawl and scrape data from the Google news aggregation service. This is a well-structured website containing time-tagged news items. All sites are different, and the specific instructions for different sites would naturally be different; the instructions in the chapter would need to be varied for these. Detailed instructions for the Google site are given, and this, of necessity, drills into detail about the structure of HTML pages and how to navigate through them. The heavy lifting is done using the Python packages “scrapy” and “BeautifulSoup”, but some details relating to use of XPath are also covered. There are many different ways to store timestamp information. This is a problem, and the chapter describes how conversion to a common format can be achieved. Visualizing results is key, and the use of the open-source SigmaJS package is described.

Chapter 9, “Text Classification Using Python”, uses Python together with a number of packages to show how these can be used to classify movie reviews using different classification models. The Natural Language Toolkit (NLTK) package provides libraries to perform various processing activities such as parsing, tokenising, and stemming of text data. This is used in conjunction with the Scikit package, which provides more advanced text processing capabilities such as TF-IDF to create word vectors from movie review data. The data set contains positive and negative reviews, and supervised models are built and their performance checked using library capabilities from the Scikit learn package. Having performed an initial basic analysis, a more sophisticated approach using word n-grams is adopted to yield improvements in performance. Further improvements are seen with the removal of stop words. The general approach taken is illustrative of the normal method adopted when performing such investigations.

Part 4 contains three chapters using R. Chapter 10, titled “Sentiment Analysis of Stock Market Behavior from Twitter Using the R Tool”, describes sentiment analysis of Twitter messages applied to the prediction of stock market behaviour. The chapter compares how well manually labelled data is predicted using various unsupervised lexical-based sentiment models or by using supervised machine learning techniques. The conclusion is that supervised techniques are superior, but in the absence of labelled training data, which is generally difficult to obtain, the unsupervised techniques have a part to play. The chapter uses R and well illustrates how most data mining is about cleaning and restructuring data. The chapter includes practical examples that are normally seen during text mining, including removal of numbers, removal of punctuation, stemming, forcing to lowercase, elimination of stop words, and pruning to remove frequent terms.

Chapter 11, titled “Topic Modeling”, relates to topic modeling as a way to understand the essential characteristics of some text data. Mining text documents usually causes vast amounts of data to be created. When representing many documents as rows, it is not unusual to have tens of thousands of dimensions corresponding to words. When considering bigrams, the number of dimensions can rise even more significantly. Such huge data sets can present considerable challenges in terms of time to process. Clearly, there is value in anything that can reduce the number of dimensions to a significantly smaller number while retaining the essential characteristics of it so that it can be used in typical data mining activities. This chapter is about topic modeling, which is one relatively new technique that shows promise to address this issue. The basic assumption behind this technique is that documents contain a probabilistic mixture of topics, and each topic itself contains a distribution of words. The generation of a document can be conceived of as the selection of a topic from one of the available ones and from there randomly select a word. Proceed word by word until the document is complete. The reverse process, namely, finding the optimum topics based on a document, is what this chapter concerns itself with. The chapter makes extensive use of R and in particular the “topicmodels” package and has ‘worked examples to allow the reader to replicate the details. As with many text mining activities, the first step is to read and preprocess the data. This involves stemming, stop word removal, removal of numbers and punctuation, and forcing to lowercase. Determination of the optimum number of topics is a trial and error process and an important consideration is the amount of pruning necessary to strike a balance between frequent and rare words. The chapter then proceeds with the detail of finding topic models, and advanced techniques are shown based on use of the topicmodels package. The determination of the optimum number of topics still requires trial and error, and visualisation approaches are shown to facilitate this.

Chapter 12 titled “Empirical Analysis of the Stack Overflow Tags Network”, presents a new angle on exploring text data using network graphs where a graph in this context means the mathematical construct of vertices connected with edges. The specific text data to be explored is from Stack Overflow. This website contains questions and answers tagged with mandatory topics. The approach within the chapter is to use the mandatory topic tags as vertices on a graph and to connect these with edges to represent whether the tags appear in the same question. The more often pairs of tags appear in questions, the larger the weight of the edge between the vertices corresponding to the tags. This seemingly simple approach leads to new insights into how tags relate to one another. The chapter uses worked R examples with the igraph package and gives a good introductory overview of some important concepts in graph exploration that this package provides. These include whether the graph is globally connected, what clusters it contains, node degree as a proxy for importance, and various clustering coefficients and path lengths to show that the graph differs from random and therefore contains significant information. The chapter goes on to

show how to reduce the graph while trying to retain interesting information and using certain node importance measures such as betweenness and closeness to give insights into tags. The interesting problem of community detection is also illustrated. Methods to visualise the data are also shown since these, too, can give new insights. The aim of the chapter is to expose the reader to the whole area of graphs and to give ideas for their use in other domains. The worked examples using Stack Overflow data serve as an easy-to-understand domain to make the explanations easier to follow.



This page intentionally left blank

---

## About the Editors

### Markus Hofmann

Dr. Markus Hofmann is currently a lecturer at the Institute of Technology Blanchardstown, Ireland, where he focuses on the areas of data mining, text mining, data exploration and visualisation, and business intelligence. He holds a PhD from Trinity College Dublin, an MSc in Computing (Information Technology for Strategic Management) from the Dublin Institute of Technology, and a BA in Information Management Systems. He has taught extensively at the undergraduate and postgraduate levels in the fields of data mining, information retrieval, text/web mining, data mining applications, data preprocessing and exploration, and databases. Dr. Hofmann has published widely at national as well as international level and specialised in recent years in the areas of data mining, learning object creation, and virtual learning environments. Further, he has strong connections to the business intelligence and data mining sectors, on both academic and industry levels. Dr. Hofmann has worked as a technology expert together with 20 different organisations in recent years for companies such as Intel. Most of his involvement was on the innovation side of technology services and for products where his contributions had significant impact on the success of such projects. He is a member of the Register of Expert Panellists of the Irish Higher Education and Training Awards council, external examiner to two other third-level institutes, and a specialist in undergraduate and postgraduate course development. He has been an internal and external examiner of postgraduate thesis submissions. He also has been a local and technical chair of national and international conferences.

### Andrew Chisholm

Andrew Chisholm holds an MA in Physics from Oxford University and over a long career has been a software developer, systems integrator, project manager, solution architect, customer-facing presales consultant, and strategic consultant. Most recently, he has been a product manager creating profitable test and measurement solutions for communication service providers. A lifelong interest in data came to fruition with the completion of a masters degree in business intelligence and data mining from the Institute of Technology, Blanchardstown, Ireland. Since then he has become a certified RapidMiner Master (with official number 7, which pads nicely to 007) and has published papers, a book chapter relating to the practical use of RapidMiner for unsupervised clustering and has authored a book titled *Exploring Data with RapidMiner*. Recently, he has collaborated with Dr. Hofmann to create both basic and advanced RapidMiner video training content for RapidMinerResources.com. In his current role, he is now combining domain knowledge of the telecommunications in-

dustry with data science principles and practical hands-on work to help customers exploit the data produced by their solutions. He fully expects data to be where the fun will be.

---

# *List of Contributors*

## Editors

- **Markus Hofmann**, Institute of Technology Blanchardstown, Ireland
- **Andrew Chisholm**, Information Gain Ltd., UK

## Chapter Authors

- **Nelson Areal**, Department of Management, University of Minho, Braga, Portugal
- **Patrick Buckley**, Institute of Technology, Blanchardstown, Ireland
- **Brian Carter**, IBM Analytics, Dublin, Ireland
- **Andrew Chisholm**, Information Gain Ltd., UK
- **David Colton**, IBM, Dublin, Ireland
- **Paul Clough**, Information School, University of Sheffield, UK
- **Paulo Cortez**, ALGORITMI Research Centre/Department of Information Systems, University of Minho, Guimarães, Portugal
- **Pavlina Davcheva**, Chair of Information Systems II, Institute of Information Systems, Friedrich-Alexander-University Erlangen-Nuremberg, Germany
- **Kyle Goslin**, Department of Computer Science, College of Computing Technology, Dublin, Ireland
- **Tobias Kötter**, KNIME.com, Berlin, Germany
- **Nuno Oliveira**, ALGORITMI Research Centre, University of Minho, Guimarães, Portugal
- **Alexander Piazza**, Chair of Information Systems II, Institute of Information Systems, Friedrich-Alexander-University Erlangen-Nuremberg, Germany
- **Tony Russell-Rose**, UX Labs, UK
- **John Ryan**, Blanchardstown Institute of Technology, Dublin, Ireland
- **Rosaria Silipo**, KNIME.com, Zurich, Switzerland
- **Kilian Thiel**, KNIME.com, Berlin, Germany
- **Christos Iraklis Tsatsoulis**, Nodalpoint Systems, Athens, Greece
- **Phil Winters**, KNIME.com, Zurich, Switzerland

This page intentionally left blank

---

## Acknowledgments

Many people have contributed to making this book and the underlying open-source software solutions a reality. We are thankful to all of you.

We would like to thank the contributing authors of this book, who shared their experience in the chapters and who thereby enable others to have a quick and successful text mining start with open-source tools, providing successful application examples and blueprints for the readers to tackle their text mining tasks and benefit from the strength of using open and freely available tools.

Many thanks to Dr. Brian Nolan, Head of School of Informatics, Institute of Technology Blanchardstown (ITB); and Dr. Anthony Keane, Head of Department of Informatics, ITB for continuously supporting projects such as this one.

Many thanks also to our families. MH: A special thanks goes to Glenda, Killian, Darragh, Daniel, SiSi, and Judy for making my life fun; My parents, Gertrud and Karl-Heinz Hofmann, for continuously supporting my endeavours. Also a huge thank you to Hans Trautwein and Heidi Krauss for introducing me to computers and my first data related application, MultiPlan, in 1986. AC: To my parents for making it possible and to my wife for keeping it possible.

The entire team of the Taylor & Francis Group was very professional, responsive, and always helpful in guiding us through this project. Should any of you readers consider publishing a book, we can highly recommend this publisher.

Open-source projects grow strong with their community. We are thankful to all contributors, particularly, text analysis — related open source-tools and all supporters of these open-source projects. We are grateful not only for source code contributions, community support in the forum, and bug reports and fixes but also for those who spread the word with their blogs, videos, and word of mouth.

With best regards and appreciation to all contributors,

**Dr. Markus Hofmann**, Institute of Technology Blanchardstown, Dublin, Ireland

**Andrew Chisholm**, Information Gain Ltd., UK

This page intentionally left blank

---

# List of Figures

1.1	Creating your repository: Overall process. . . . .	6
1.2	Creating your repository: Step A – Get Page operator. . . . .	7
1.3	Creating your repository: Step B. . . . .	8
1.4	Creating your repository: Step B – vector window. . . . .	8
1.5	Creating your repository: Step B – Cut Document operator. . . . .	9
1.6	Creating your repository: Step B – Extract Information operator. . . . .	9
1.7	Creating your repository: Step C – create attributes. . . . .	9
1.8	Creating your repository: Step C – attribute name. . . . .	10
1.9	Creating your repository: Step D. . . . .	10
1.10	Creating your repository: Step E – Write Excel operator. . . . .	10
1.11	Build a token repository: Process I. . . . .	11
1.12	Build a token repository: Process I Step A – Read Excel operator. . . . .	12
1.13	Build a token repository: Process I Step B – Get Pages operator. . . . .	12
1.14	Build a token repository: Process I Step C – Process Documents operator. . . . .	13
1.15	Build a token repository: Process I Step C – vector window. . . . .	13
1.16	Build a token repository: Process I Step C – Extract Information operator. . . . .	13
1.17	Build a token repository: Process I Step C – extracting date. . . . .	14
1.18	Build a token repository: Process I Step C – extracting president’s name. . . . .	14
1.19	Build a token repository: Process I Step C – extracting regular expression. . . . .	15
1.20	Build a token repository: Process I Step C – regular region. . . . .	16
1.21	Build a token repository: Process I Step C – cutting speech content. . . . .	16
1.22	Build a token repository: Process I Step C – cut document nodes. . . . .	16



1.23 Build a token repository: Process I Step D – store repository. . . . . 16

1.24 Build a token repository: Process II. . . . . 17

1.25 Build a token repository: Process II Step A – Retrieve operator. . . . . 17

1.26 Build a token repository: Process II Step B – Rename operator. . . . . 18

1.27 Build a token repository: Process II Step B – additional attributes. . . . . 18

1.28 Build a token repository: Process II Step C – Select Attributes operator. . 18

1.29 Build a token repository: Process II Step C – subset attributes. . . . . 19

1.30 Build a token repository: Process II Step D – Write Database operator. . . 19

1.31 Analyzing the corpus: Process I. . . . . 20

1.32 Analyzing the corpus: Process I Step A – Read Database operator. . . . . 21

1.33 Analyzing the corpus: Process I Step A – SQL. . . . . 21

1.34 Analyzing the corpus: Process I Step B – term occurrences. . . . . 22

1.35 Analyzing the corpus: Process I Step B – vector creation window. . . . . 22

1.36 Analyzing the corpus: Process I Step B – Extract Content operator. . . . . 23

1.37 Analyzing the corpus: Process I Step B – Tokenize operator. . . . . 23

1.38 Analyzing the corpus: Process I Step B – Filter Stopwords (English)  
operator. . . . . 24

1.39 Analyzing the corpus: Process I Step B – custom dictionary. . . . . 24

1.40 Analyzing the corpus: Process I Step C – Write Excel operator. . . . . 24

1.41 Analyzing the corpus: Process I Step C – transposed report. . . . . 25

1.42 Analyzing the corpus: Process II Step B. . . . . 26

1.43 Analyzing the corpus: Process II Step B – Generate n-Grams (Terms)  
operator. . . . . 26

1.44 Analyzing the corpus: Process II Step B – Filter Tokens (by Content)  
operator. . . . . 26

1.45 Analyzing the corpus: Process II Step C. . . . . 26

1.46 Visualization: Layout of transposed worksheet. . . . . 28

1.47 Visualization: Wordle menu. . . . . 29

1.48 Visualization: Copy and paste data into create section. . . . . 30

1.49	Visualization: Speech represented as a word cloud. . . . .	30
1.50	Visualization: Word cloud layout. . . . .	31
1.51	Visualization: Word cloud colour. . . . .	31
1.52	Visualization: Word cloud token limit. . . . .	32
1.53	Visualization: Word cloud font types. . . . .	32
1.54	Visualization: Word cloud remove tokens (filtered to 20 tokens). . . . .	33
1.55	Visualization: Word cloud filter option (remove token). . . . .	33
1.56	Visualization: Word cloud filtered (specific tokens removed). . . . .	34
1.57	Visualization: Word cloud options (print, save, new window, randomize). . . . .	34
1.58	Visualization: Layout of transposed bigram worksheet. . . . .	34
1.59	Visualization: Word cloud bigrams. . . . .	35
1.60	Visualization: Word cloud bigrams filtered (removal of tokens). . . . .	35
2.1	Observed variation for the word “the” for consecutive 5,000-word windows within the novel <i>Moby Dick</i> . . . . .	42
2.2	RapidMiner process to calculate word frequencies. . . . .	43
2.3	Process Section A within Figure 2.2. . . . .	44
2.4	Process Section B within Figure 2.2. . . . .	44
2.5	Process Section C within Figure 2.2. . . . .	45
2.6	Process Section D within Figure 2.2. . . . .	45
2.7	RapidMiner process to execute process for all attributes to fit Zipf-Mandelbrot distribution. . . . .	48
2.8	Detail of RapidMiner process to execute Zipf-Mandelbrot distribution fit. . . . .	48
2.9	RapidMiner process to fit Zipf-Mandelbrot distribution. . . . .	49
2.10	Configuration for <i>Optimize Parameters (Evolutionary)</i> operator. . . . .	50
2.11	Details for <i>Optimize Parameters (Evolutionary)</i> operator. . . . .	50
2.12	Details for macro-generation workaround to pass numerical parameters to Optimize Parameters operator. . . . .	51
2.13	Calculation of Zipf-Mandelbrot probability and error from known probability. . . . .	51