
Contents

I	RapidMiner	1
1	RapidMiner for Text Analytic Fundamentals	3
	<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
2	Empirical Zipf-Mandelbrot Variation for Sequential Windows within Documents	37
	<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	KNIME	61
3	Introduction to the KNIME Text Processing Extension	63
	<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	Social Media Analysis — Text Mining Meets Network Mining	81
	<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

III	Python	93
5	Mining Unstructured User Reviews with Python	95
	<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
6	Sentiment Classification and Visualization of Product Review Data	133
	<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
7	Mining Search Logs for Usage Patterns	153
	<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

8	Temporally Aware Online News Mining and Visualization with Python	173
	<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
9	Text Classification Using Python	199
	<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	R	295
-------	---	-----