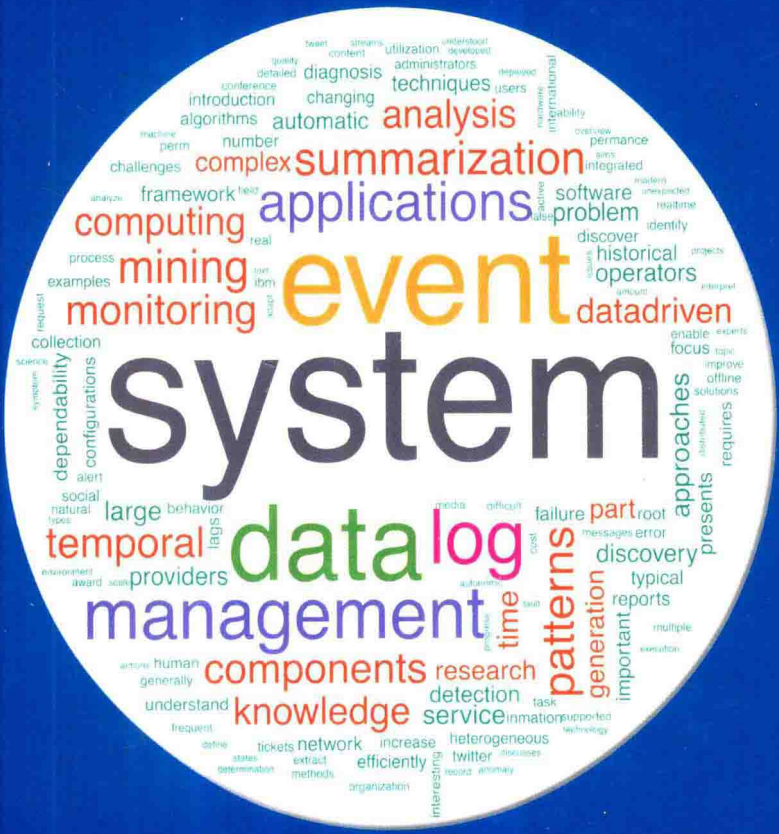
Chapman & Hall/CRC

Data Mining and Knowledge Discovery Series

EVENT MINING

ALGORITHMS AND APPLICATIONS



Edited by

Tao Li

CRC Press
Taylor & Francis Group

A CHAPMAN & HALL BOOK

EVENT MINING
ALGORITHMS AND APPLICATIONS

Edited by
Tao Li

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

Printed at CPI UK on sustainably sourced paper
Version Date: 20150729

International Standard Book Number-13: 978-1-4665-6857-0 (Hardback)

EVENT MINING
ALGORITHMS AND APPLICATIONS

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

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

THE TOP TEN ALGORITHMS IN DATA MINING
Xindong Wu and Vipin Kumar

UNDERSTANDING COMPLEX DATASETS: DATA MINING WITH MATRIX DECOMPOSITIONS
David Skillicorn

To the School of Computing and Information Sciences (SCIS) at Florida International University (FIU)
and
To the School of Computer Science at Nanjing University of Posts and Telecommunications (NJUPT)

Preface

Many systems, from computing systems, physical systems, business systems, to social systems, are only observable indirectly from the events they emit. Events can be defined as real-world occurrences and they typically involve changes of system states. Events are naturally temporal and are often stored as logs, e.g., business transaction logs, stock trading logs, sensor logs, computer system logs, HTTP requests, database queries, network traffic data, etc. These events capture system states and activities over time. For effective system management, a system needs to automatically monitor, characterize, and understand its behavior and dynamics, mine events to uncover useful patterns, and acquire the needed knowledge from historical log/event data.

Event mining is a series of techniques for automatically and efficiently extracting valuable knowledge from historical event/log data and plays an important role in system management. The purpose of this book is to present a variety of event mining approaches and applications with a focus on computing system management. It is mainly intended for researchers, practitioners, and graduate students who are interested in learning about the state of the art in event mining. It can also serve as a textbook for advanced courses. Learning about event mining is challenging as it is an inter-disciplinary field that requires familiarity with several research areas and the relevant literature is scattered in a variety of publication venues such as the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD), IEEE International Conference in Data Mining (IEEE ICDM), IEEE/IFIP Network Operations and Management Symposium (NOMS), International Conference on Network and Service Management (CNSM), and IFIP/IEEE Symposium on Integrated Network and Service Management (IM). We hope that this book will make the field easier to approach by providing a good starting point for readers not familiar with the topic as well as a comprehensive reference for those working in the field.

Although the chapters of the book are mostly self-contained and can be read in any order, they have been grouped and ordered in a way that can provide a structured introduction to the topic. In particular, after Chapter 1 (Introduction), the book is organized as follows:

Part I: Event Generation and System Monitoring

- Chapter 2: Event Generation: From Logs to Events
- Chapter 3: Optimizing System Monitoring Configurations

Part II: Event Pattern Discovery and Summarization

- Chapter 4: Event Pattern Mining
- Chapter 5: Mining Time Lags
- Chapter 6: Log Event Summarization

Part III: Applications

- Chapter 7: Data-Driven Applications in System Management
- Chapter 8: Social Media Event Summarization Using Twitter Streams

I would like to thank Dr. Sheng Ma, Dr. Charles Perng, Dr. Larisa Shwartz, and Dr. Genady Grabarnik for their long-term research collaboration on event mining. The research studies presented in the book are based on the research projects conducted at the Knowledge Discovery Research Group (KDRG) in the School of Computing and Information Sciences (SCIS) at Florida International University (FIU). The research projects have been partially supported by the National Science Foundation (NSF)(NSF CAREER Award IIS-0546280, CCF-0830659, HRD-0833093, DMS-0915110, CNS-1126619, and IIS-1213026), the U.S. Department of Homeland Security under grant award number 2010-ST-062-00039, the Army Research Office under grant number W911NF-10-1-0366 and W911NF-12-1-0431, a 2005 IBM Shared University Research (SUR) Award, and IBM Faculty Research Awards (2005, 2007, and 2008). The research projects have also been supported by Florida International University (FIU), Nanjing University of Posts and Telecommunications (NJUPT), Xiamen University (XMU), Nanjing University of Science and Technology (NJUST), and Xiamen University of Technology (XMUT).

Editing a book takes a lot of effort. I would like to thank the following members of the Knowledge Discovery Research Group (KDRG) in the School of Computing and Information Sciences (SCIS) at Florida International University (FIU) for the contributions of their chapters as well as their help in reviewing and proofreading:

- Dr. Yexi Jiang (now works at Facebook Inc.)
- Dr. Chao Shen (now works at Amazon Inc.)
- Dr. Liang Tang (now works at LinkedIn Inc.)
- Chunqiu Zeng
- Wubai Zhou

I would also like to thank the KDRG group members (Wei Liu, Ming Ni, Bin Xia, Jian Xu, Wei Xue, and Longhui Zhang) for proofreading the book and for their valuable suggestions and comments. I would also like to thank the people at Chapman & Hall/Taylor & Francis for their help and encouragement.

Tao Li

List of Figures

1.1 The architecture of an integrated data-driven system management framework. (See color insert.) 2

2.1 Event timeline for the FileZilla log example. 15
2.2 An example of log generation code in [232]. 19
2.3 An example of log message classification. 20
2.4 An example of hierarchal log message classification. 20
2.5 An example of word simple match based similarity. 22
2.6 Two status messages in PVFS2. 25
2.7 A case study of the Apache HTTP server log. (See color insert.) 31
2.8 Function $g(r)$, $|C| = 100$. 40
2.9 Vocabulary size. 41
2.10 Varying parameter λ'. 45

3.1 Relationship of monitoring software, administrators, and customer servers. 50
3.2 Portal of IBM Tivoli monitoring. (See color insert.) 52
3.3 False positive alert duration. 54
3.4 Flowchart for ticket creation. 55
3.5 Number of situation issues in two months of manual tickets. 58
3.6 Flow chart of classification model. 60
3.7 Eliminated false positive tickets. 62
3.8 Postponed real tickets. 62
3.9 Comparison with revalidate method. 63
3.10 Accuracy of situation discovery for file system space alert. . 64
3.11 Accuracy of situation discovery for disk space alert. 64
3.12 Accuracy of situation discovery for service not available. . . 65
3.13 Accuracy of situation discovery for router/switch down. . . 65

4.1 Temporal data in system management. (See color insert.) . . 72
4.2 Social events in social media. (See color insert.) 73
4.3 A sequential pattern example. 74
4.4 Event sequences are illustrated with timestamps. 84
4.5 Steps of p-pattern discovery. 87

4.6 T-patterns are qualified by a statistical hypothesis test. (See color insert.) . . . 93
4.7 Episodes of event sequences. . . . 96
4.8 There are three types of episodes: (a) parallel episode; (b) serial episode; (c) composite episode. The vertical box indicates that all the event types in the box happen in a single episode, but the order of events does not matter. The arrow means the order of events. . . . 97
4.9 Two states q_0 and q_1 correspond to the states with a low rate and a high rate, respectively. p denotes the probability of state change. The events generated in both states q_0 and q_1 are mixed into an event sequence. . . . 99
4.10 A fixed window size 6 is given. Predicting rare event F is transformed to searching for the frequent patterns preceding event F. (See color insert.) . . . 101
4.11 The CPU usage is continuous time series data, while the system events are identified when starting different types of tasks such as disk intensive task and CPU intensive task. . . . 103
4.12 The front sub-series is a snippet of time series with a fixed time window size before the occurrence of an event, while the rear sub-series is a snippet of time series with the same fixed time window size after the occurrence of the event. . . . 104
4.13 A MapQL query on real property data is given, where POINT (-80.27,25.757228) is the location of Florida International University. . . . 106
4.14 The MapQL query result on real property data is displayed on the map. . . . 107
4.15 System overview. . . . 108
4.16 The syntax tree for a MapQL statement "SELECT geo FROM street WHERE name LIKE 'sw 8th';". . . . 112
4.17 An example illustrating the PrefixSpan algorithm. . . . 113
4.18 Example of a generated template. . . . 114
4.19 Node types in a workflow. . . . 115
4.20 Workflow examples. . . . 116
4.21 The workflow of searching for house properties with a good appreciation potential. All the sub-tasks in the workflow are scheduled by FIU-Miner and are executed in the distributed environment. . . . 117
4.22 Average property prices by zip code in Miami. (See color insert.) . . . 118
4.23 Detailed properties in Miami. . . . 118
4.24 A template for searching the neighborhood, given the partial name of the street. . . . 119
4.25 Final analysis results. (See color insert.) . . . 119

5.1 The temporal dependencies between A and B are denoted as direct edges. 124
5.2 Lag interval for temporal dependency. 126
5.3 A sorted table. 128
5.4 Incremental sorted table. 132
5.5 The j^{th} event B occurring at b_j can be implied by any event A. Variable $z_{ij} = 1$ if the j^{th} event B is associated with i^{th} event A, and 0 otherwise. 134
5.6 $A \rightarrow_L B$, where $L \sim \mathcal{N}(\mu, \sigma^2)$. An event A that occurred at time a_i is associated with an event B that occurred at b_j with probability $\mathcal{N}(b_j - a_i|\mu, \sigma^2)$. Here μ is the expected time lag of an occurrence of B after a_i. (see color insert.) 137
5.7 The KL distance between the ground truth and the one learned by each algorithm. 143
5.8 Time cost comparison. ϵ of *appLagEM* is set to 0.001, 0.05, 0.1, 0.2, 0.4, and 0.8. The sizes of the datasets range from 200 to $40k$. 144
5.9 The overview of the temporal dependency mining system (TDMS). (See color insert.) 147
5.10 Temporal dependencies are shown in a graph where each node denotes an event, and an edge between two nodes indicates the corresponding two events are dependent. (See color insert.) . 148
5.11 Temporal dependencies are discovered by setting the SNR threshold and are displayed in a table. The number of temporal dependencies depends on the SNR setting. (See color insert.) 148

6.1 An example event sequence. 158
6.2 Event summarization result produced by the solution of [119]. 159
6.3 A high level overview of summary. 159
6.4 The corresponding matrix for the event sequence in Example 6.1. 159
6.5 An example of summarizing events with HMM. 164
6.6 An example for *natural event summarization* [112]. 167
6.7 The *natural event summarization* framework proposed in [112]. 167
6.8 Inter-arrival histogram $h_{aa}(S)$ (left) and inter-arrival histogram $h_{ab}(S)$ (right). 169
6.9 Standard histogram that best approximates $h_{aa}(S)$ (left) and $h_{ab}(S)$ (right). 170
6.10 An example histogram graph. 173
6.11 An example ERN graph. 174
6.12 Segments information in a wavelet spectrum sequence. . . . 177

6.13 Before pruning, the histogram graph contains 17 vertices and 136 edges; after pruning, the histogram graph contains only 6 vertices and 14 edges. 177
6.14 An example summarization result for the security log. . . . 178
6.15 Summarization workflows of example scenarios. 181
6.16 Converting the original event sequence to the vectors. 182
6.17 Relationship between vector and ST. 184
6.18 Summarizing with META. 190

7.1 Problem detection, determination, and resolution. (See color insert.) 198
7.2 An example of LSH-DOC. 203
7.3 An example of LSH-SEP. 204
7.4 An example of $l < |Q|$. 204
7.5 Dissimilar events in segments. 209
7.6 Random sequence mask. 210
7.7 Average search cost curve ($n = 100K, |\mathcal{Z}_{H,S}| = 16, \theta = 0.5, |Q| = 10, \delta = 0.8, k = 2$). 213
7.8 RecallRatio comparison for ThunderBird log. 216
7.9 RecallRatio comparison for Apache log. 217
7.10 Number of probed candidates for ThunderBird log. 218
7.11 Number of probed candidates for Apache log. 219
7.12 RecallRatio for TG1. 219
7.13 Varying m. 220
7.14 Varying r. 220
7.15 Peak memory cost for ThunderBird log. 221
7.16 Peak memory cost for Apache log. 221
7.17 Indexing time for ThunderBird log. 222
7.18 Indexing time for Apache log. 222
7.19 A hierarchical multi-label classification problem in the IT environment. A ticket instance is shown in (a). (b) The ground truth for the ticket with multiple class labels. (c), (d), (e), and (f) Four cases with misclassification. Assuming the cost of each wrong class label is 1, Zero-one loss, Hamming loss, H-loss, HMC-loss are given for misclassification. Notably, to calculate the HMC-loss, the cost weights for FN and FP are a and b respectively. The misclassified nodes are marked with a red square. The contextual misclassification information is indicated by the green rectangle. 224
7.20 Four cases of contextual misclassification are shown in (a–d) for node i. Here the left pair is the ground truth; the right pair is the prediction. The misclassified nodes are marked with a red square. 226

7.21 Figure illustrates a hierarchy with nine nodes and the steps of algorithm 6. Nodes labeled positive are green. A dotted ellipse marks a super node composed of the nodes in it. 231
7.22 Experiments involving tickets. 233
7.23 Numbers of tickets and distinct resolutions. 237
7.24 Top repeated resolutions for event tickets. 237
7.25 Test results for $K = 10$, $k = 3$. 246
7.26 Test results for $K = 20$, $k = 5$. 246
7.27 Average penalty for varying K and k. 247
7.28 Weighted accuracy by varying k, $K = 10$. 248
7.29 Average penalty by varying k, $K = 10$. 248
7.30 Overall score by varying k, $K = 10$. 249
7.31 Overall score for varying K and k. 249
7.32 Accuracy varies for different *numTopics* for dataset account4. 252
7.33 Test results for three accounts by varying k for $K = 8$. . . . 252
7.34 Test results for three accounts by varying k for $K = 16$. . . 253
7.35 Similarity measure before and after metric learning for a training set. (See color insert.) 253
7.36 Similarity measure before and after metric learning for a testing set. (See color insert.) 254
7.37 Mean average precision (MAP) varying parameter K of the underlying KNN algorithm. 254

8.1 Plate notation of the mixture model. 268
8.2 Precision of participant detection performance on phrase level. 276
8.3 Recall of participant detection performance on phrase level. 277
8.4 Precision of participant detection performance on participant level. 277
8.5 Recall of participant detection performance on participant level. 277
8.6 Sub-event detection performance of different methods without participant detection. 278
8.7 Sub-event detection performance of different methods with participant detection. 278
8.8 Participant detection performance on phrase level. 280

List of Tables

2.1 An example of FileZilla's log 15
2.2 Summary of the three types of approaches 17
2.3 Experimental machines 27
2.4 Log data summary 27
2.5 Summary of comparative methods 28
2.6 F-Measures of K-Medoids 29
2.7 F-Measures of Single-Linkage 29
2.8 Parameter settings 30
2.9 Example of match score 33
2.10 Example of two message groups 37
2.11 Experimental machine 40
2.12 Summary of collected system logs 41
2.13 Summary of comparative algorithms 42
2.14 Summary of small log data 43
2.15 Average F-measure comparison 43
2.16 Discovered message signatures 44

3.1 Definitions for alert, event, and ticket 53
3.2 Domain word examples 59
3.3 Data summary 61
3.4 Sampled rules for Account2 with testing data ratio = 0.3 . . 63
3.5 Accuracy of the word-match method 66

4.1 An example of a sequence database 75
4.2 Candidate generation in GSP 77
4.3 Example for SPADE 77
4.4 An example of the FreeSpan algorithm 78
4.5 Projected database of $< a >$ in PrefixSpan 80
4.6 An example to illustrate m-patterns 89
4.7 Summary of mining event patterns 105
4.8 A snippet of MapQL logs 111

5.1 Temporal patterns with time lag 124
5.2 Parameters for synthetic data generation 141
5.3 Experimental results on synthetic datasets. 142
5.4 Snippet of discovered time lags 143

5.5 Real event dataset 145

6.1 Distinction between event pattern mining and event summarization 154
6.2 Experiments with real *Windows event log* datasets (results are obtained from [119]) 163
6.3 Occurrences of event types E_a and E_b 169
6.4 Experimental evaluation results 178
6.5 A brief summary of the event summarization methods . . . 179
6.6 Notations of basic operations 188

7.1 Definitions for alert, event, and ticket 199
7.2 An example of a hash value table 205
7.3 Hash value sequence $h_i(S)$ 206
7.4 Sorted suffixes of $h_i(S)$ 208
7.5 Experimental machine 214
7.6 Testing query groups 214
7.7 Number of true results 215
7.8 "SuffixMatrix(Strict)" for TG1 217
7.9 Special cases of CH-loss 228
7.10 Comparison with the "Flat" classification 235
7.11 Data summary 236
7.12 Notations 238
7.13 Event attribute types 238
7.14 Tickets for explaining motivation of incorporating resolution information 243
7.15 First six words are extracted to represent topics trained from LDA 243
7.16 Three resolution types with the event description they resolved . 251

8.1 Statistics of the datasets, including five NBA basketball games and the WWDC 2012 conference event 272
8.2 Example participants for the NBA game Spurs vs Okc (2012-5-31) and the WWDC'12 conference 273
8.3 An example clip of the play-by-play of an NBA game (*heatvsokc*) 274
8.4 An example clip of the live updates of WWDC 2012 275
8.5 Summary of comparative algorithms 279
8.6 F-1 score of ROUGE-2 of event summarization using different sub-event detection methods 280