Collective Intelligence in Action

SATNAM ALAG
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>foreword</td>
<td>xv</td>
</tr>
<tr>
<td>preface</td>
<td>xvii</td>
</tr>
<tr>
<td>acknowledgments</td>
<td>xix</td>
</tr>
<tr>
<td>about this book</td>
<td>xxi</td>
</tr>
</tbody>
</table>

## Part 1  Gathering Data for Intelligence

### 1 Understanding collective intelligence

1. What is collective intelligence? 4
2. CI in web applications 6
   - Collective intelligence from the ground up: a sample application 7
   - Benefits of collective intelligence 9
   - CI is the core component of Web 2.0 10
   - Harnessing CI to transform from content-centric to user-centric applications 12
3. Classifying intelligence 14
   - Explicit intelligence 14
   - Implicit intelligence 15
   - Derived intelligence 16
4. Summary 18
5. Resources 18
2 Learning from user interactions 20

2.1 Architecture for applying intelligence 21
Synchronous and asynchronous services 21 • Real-time learning in an event-driven system 23 • Polling services for non-event-driven systems 24 • Advantages and disadvantages of event-based and non-event-based architectures 25

2.2 Basics of algorithms for applying CI 25
Users and items 26 • Representing user information 27 • Content-based analysis and collaborative filtering 29 • Representing intelligence from unstructured text 30 • Computing similarities 31 • Types of datasets 32

2.3 Forms of user interaction 34
Rating and voting 35 • Emailing or forwarding a link 36 • Bookmarking and saving 36 • Purchasing items 37 • Click-stream 37 • Reviews 39

2.4 Converting user interaction into collective intelligence 41
Intelligence from ratings via an example 41 • Intelligence from bookmarking, saving, purchasing items, forwarding, click-stream, and reviews 46

2.5 Summary 48

2.6 Resources 48

3 Extracting intelligence from tags 50

3.1 Introduction to tagging 51
Tag-related metadata for users and items 52 • Professionally generated tags 52 • User-generated tags 53 • Machine-generated tags 54 • Tips on tagging 55 • Why do users tag? 55

3.2 How to leverage tags 56
Building dynamic navigation 56 • Innovative uses of tag clouds 58 • Targeted search 59 • Folksonomies and building a dictionary 60

3.3 Extracting intelligence from user tagging: an example 60
Items related to other items 61 • Items of interest for a user 61 • Relevant users for an item 62

3.4 Scalable persistence architecture for tagging 62
Reviewing other approaches 63 • Recommended persistence architecture 66

3.5 Building tag clouds 69
Persistence design for tag clouds 69 • Algorithm for building a tag cloud 70 • Implementing a tag cloud 71 • Visualizing a tag cloud 76
CONTENTS

6 Intelligent web crawling 145

6.1 Introducing web crawling 146

Why crawl the Web? 146 • The crawling process 147
Intelligent crawling and focused crawling 149 • Deep
crawling 150 • Available crawlers 151

6.2 Building an intelligent crawler step by step 152

Implementing the core algorithm 152 • Being polite: following the
robots.txt file 156 • Retrieving the content 159 • Extracting
URLs 160 • Making the crawler intelligent 161 • Running the
crawler 162 • Extending the crawler 163

6.3 Scalable crawling with Nutch 164

Setting up Nutch 164 • Running the Nutch crawler 165 • Searching
with Nutch 168 • Apache Hadoop, MapReduce, and Dryad 169

6.4 Summary 171

6.5 Resources 171

PART 2 DERIVING INTELLIGENCE ......................... 173

7 Data mining: process, toolkits, and standards 175

7.1 Core concepts of data mining 176

Attributes 176 • Supervised and unsupervised learning 178
Key learning algorithms 178 • The mining process 181

7.2 Using an open source data mining framework: WEKA 182

Using the WEKA application: a step-by-step tutorial 183
Understanding the WEKA APIs 186 • Using the WEKA APIs
via an example 188

7.3 Standard data mining API: Java Data Mining (JDM) 193

JDM architecture 194 • Key JDM objects 195 • Representing the
dataset 196 • Learning models 197 • Algorithm settings 199
JDM tasks 199 • JDM connection 200 • Sample code for accessing
DME 202 • JDM models and PMML 204

7.4 Summary 204

7.5 Resources 205

8 Building a text analysis toolkit 206

8.1 Building the text analyzers 207

Leveraging Lucene 208 • Writing a stemmer analyzer 213 • Writing
a TokenFilter to inject synonyms and detect phrases 214 • Writing an
analyzer to inject synonyms and detect phrases 218 • Putting our
analyzers to work 218
CONTENTS

8.2 Building the text analysis infrastructure 221

Building the tag infrastructure 222 • Building the term vector infrastructure 225 • Building the Text Analyzer class 231

Applying the text analysis infrastructure 234

8.3 Use cases for applying the framework 237

8.4 Summary 238

8.5 Resources 239

9 Discovering patterns with clustering 240

9.1 Clustering blog entries 241

Defining the text clustering infrastructure 242 • Retrieving blog entries from Technorati 244 • Implementing the k-means algorithms for text processing 247 • Implementing hierarchical clustering algorithms for text processing 253 • Expectation maximization and other examples of clustering high-dimension sparse data 261

9.2 Leveraging WEKA for clustering 262

Creating the learning dataset 263 • Creating the clusterer 265 • Evaluating the clustering results 266

9.3 Clustering using the JDM APIs 268

Key JDM clustering-related classes 268 • Clustering settings using the JDM APIs 269 • Creating the clustering task using the JDM APIs 271 • Executing the clustering task using the JDM APIs 271 • Retrieving the clustering model using the JDM APIs 272

9.4 Summary 272

9.5 Resources 273

10 Making predictions 274

10.1 Classification fundamentals 275

Learning decision trees by example 275 • Naïve Bayes’ classifier 281 • Belief networks 285

10.2 Classifying blog entries using WEKA APIs 287

Building the dataset for classifying blog entries 288 • Building the classifier class 292

10.3 Regression fundamentals 294

Linear regression 295 • Multi-layer perceptron (MLP) 297 • Radial basis functions (RBF) 298

10.4 Regression using WEKA 299
10.5 Classification and regression using JDM 300

Key JDM supervised learning–related classes 300 • Supervised learning settings using the JDM APIs 302 • Creating the classification task using the JDM APIs 304 • Executing the classification task using the JDM APIs 304 • Retrieving the classification model using the JDM APIs 305 • Retrieving the classification model using the JDM APIs 305

10.6 Summary 306
10.7 Resources 306

PART 3 APPLYING INTELLIGENCE IN YOUR APPLICATION .....307

11 Intelligent search 309

11.1 Search fundamentals 310

Search architecture 310 • Core Lucene classes 311 • Basic indexing and searching via example 313

11.2 Indexing with Lucene 320

Understanding the index format 320 • Modifying the index 321 • Incremental indexing 322 • Accessing the term frequency vector 324 • Optimizing indexing performance 325

11.3 Searching with Lucene 327

Understanding Lucene scoring 327 • Querying Lucene 330 • Sorting search results 331 • Querying on multiple fields 333 • Filtering 334 • Searching multiple indexes 335 • Using a HitCollector 335 • Optimizing search performance 338

11.4 Useful tools and frameworks 339

Luke 339 • Solr 339 • Compass 341 • Hibernate search 341

11.5 Approaches to intelligent search 341

Augmenting search with classifiers and predictors 342 • Clustering search results 342 • Personalizing results for the user 344 • Community-based search 344 • Linguistic-based search 345 • Data search 345

11.6 Summary 347
11.7 Resources 347

12 Building a recommendation engine 349

12.1 Recommendation engine fundamentals 350

Introducing the recommendation engine 351 • Item-based and user-based analysis 352 • Computing similarity using content-based and collaborative techniques 353 • Comparison of content-based and collaborative techniques 354
12.2 Content-based analysis 355
Finding similar items using a search engine (Lucene) 355
Building a content-based recommendation engine 359 • Related items for document clusters 362 • Personalizing content for a user 362

12.3 Collaborative filtering 363
k-nearest neighbor 363 • Packages for implementing collaborative filtering 365 • Dimensionality reduction with latent semantic indexing 369 • Implementing dimensionality reduction 370 • Probabilistic model–based approach 373

12.4 Real-world solutions 373
Amazon item-to-item recommendation 374 • Google News personalization 377 • Netflix and the BellKor Solution for the Netflix Prize 381

12.5 Summary 385
12.6 Resources 386

index 389