Taming Text

HOW TO FIND, ORGANIZE, AND MANIPULATE IT

GRANT S. INGERSOLL
THOMAS S. MORTON
ANDREW L. FARRIS

MANNING
SHELTER ISLAND
Contents

foreword xiii
preface xiv
acknowledgments xvii
about this book xix
about the cover illustration xxii

1 Getting started taming text 1
1.1 Why taming text is important 2
1.2 Preview: A fact-based question answering system 4
  Hello, Dr. Frankenstein 5
1.3 Understanding text is hard 8
1.4 Text, tamed 10
1.5 Text and the intelligent app: search and beyond 11
  Searching and matching 12 • Extracting information 13
  Grouping information 13 • An intelligent application 14
1.6 Summary 14
1.7 Resources 14

2 Foundations of taming text 16
2.1 Foundations of language 17
  Words and their categories 18 • Phrases and clauses 19
  Morphology 20
CONTENTS

4 Fuzzy string matching 84

4.1 Approaches to fuzzy string matching 86

Character overlap measures 86 • Edit distance measures 89
N-gram edit distance 92

4.2 Finding fuzzy string matches 94

Using prefixes for matching with Solr 94 • Using a trie for
prefix matching 95 • Using n-grams for matching 99

4.3 Building fuzzy string matching applications 100

Adding type-ahead to search 101 • Query spell-checking for
search 105 • Record matching 109

4.4 Summary 114

4.5 Resources 114

5 Identifying people, places, and things 115

5.1 Approaches to named-entity recognition 117

Using rules to identify names 117 • Using statistical
classifiers to identify names 118

5.2 Basic entity identification with OpenNLP 119

Finding names with OpenNLP 120 • Interpreting names
identified by OpenNLP 121 • Filtering names based on
probability 122

5.3 In-depth entity identification with OpenNLP 123

Identifying multiple entity types with OpenNLP 123
Under the hood: how OpenNLP identifies names 126

5.4 Performance of OpenNLP 128

Quality of results 129 • Runtime performance 130
Memory usage in OpenNLP 131

5.5 Customizing OpenNLP entity identification
for a new domain 132

The whys and hows of training a model 132 • Training
an OpenNLP model 133 • Altering modeling inputs 134
A new way to model names 136

5.6 Summary 138

5.7 Further reading 139
CONTENTS

6 Clustering text 140

6.1 Google News document clustering 141
6.2 Clustering foundations 142

Three types of text to cluster 142 • Choosing a clustering algorithm 144 • Determining similarity 145 • Labeling the results 146 • How to evaluate clustering results 147

6.3 Setting up a simple clustering application 149
6.4 Clustering search results using Carrot2 149

Using the Carrot2 API 150 • Clustering Solr search results using Carrot2 151

6.5 Clustering document collections with Apache Mahout 154

Preparing the data for clustering 155 • K-Means clustering 158

6.6 Topic modeling using Apache Mahout 162
6.7 Examining clustering performance 164

Feature selection and reduction 164 • Carrot2 performance and quality 167 • Mahout clustering benchmarks 168

6.8 Acknowledgments 172
6.9 Summary 173
6.10 References 173

7 Classification, categorization, and tagging 175

7.1 Introduction to classification and categorization 177
7.2 The classification process 180

Choosing a classification scheme 181 • Identifying features for text categorization 182 • The importance of training data 183 • Evaluating classifier performance 186 • Deploying a classifier into production 188

7.3 Building document categorizers using Apache Lucene 189

Categorizing text with Lucene 189 • Preparing the training data for the MoreLikeThis categorizer 191 • Training the MoreLikeThis categorizer 193 • Categorizing documents with the MoreLikeThis categorizer 197 • Testing the MoreLikeThis categorizer 199 • MoreLikeThis in production 201
7.4 Training a naive Bayes classifier using Apache Mahout

Categorizing text using naive Bayes classification 202
Preparing the training data 204 • Withholding test data 207
Training the classifier 208 • Testing the classifier 209
Improving the bootstrapping process 210 • Integrating the Mahout Bayes classifier with Solr 212

7.5 Categorizing documents with OpenNLP 215

Regression models and maximum entropy • document categorization 216 • Preparing training data for the maximum entropy document categorizer 219 • Training the maximum entropy document categorizer 220 • Testing the maximum entropy document classifier 224 • Maximum entropy document categorization in production 225

7.6 Building a tag recommender using Apache Solr 227

Collecting training data for tag recommendations 229
Preparing the training data 231 • Training the Solr tag recommender 232 • Creating tag recommendations 234
Evaluating the tag recommender 236

7.7 Summary 238
7.8 References 239

8 Building an example question answering system 240

8.1 Basics of a question answering system 242
8.2 Installing and running the QA code 243
8.3 A sample question answering architecture 245
8.4 Understanding questions and producing answers 248

Training the answer type classifier 248 • Chunking the query 251 • Computing the answer type 252 • Generating the query 255 • Ranking candidate passages 256

8.5 Steps to improve the system 258
8.6 Summary 259
8.7 Resources 259

9 Untamed text: exploring the next frontier 260

9.1 Semantics, discourse, and pragmatics: exploring higher levels of NLP 261

Semantics 262 • Discourse 263 • Pragmatics 264