An Introduction to Information Retrieval

Christopher D. Manning
Prabhakar Raghavan
Hinrich Schütze

Cambridge University Press
Cambridge, England

Preliminary draft (c) 2007 Cambridge UP
Brief Contents

1 Information retrieval using the Boolean model 1
2 The dictionary and postings lists 17
3 Tolerant retrieval 39
4 Index construction 51
5 Index compression 65
6 Scoring and term weighting 85
7 Vector space retrieval 97
8 Evaluation in information retrieval 111
9 Relevance feedback and query expansion 133
10 XML retrieval 151
11 Probabilistic information retrieval 167
12 Language models for information retrieval 181
13 Text classification and Naïve Bayes 193
14 Vector space classification 217
15 Support vector machines and kernel functions 237
16 Flat clustering 251
17 Hierarchical clustering 275
18 Dimensionality reduction and Latent Semantic Indexing 297
19 Web search basics 307
20 Web crawling and indexes 323
21 Link analysis 337
Contents

List of Tables xv
List of Figures xvii
Table of Notations xxiii
Preface xxv

1 Information retrieval using the Boolean model 1
  1.1 An example information retrieval problem 2
  1.2 A first take at building an inverted index 5
  1.3 Processing Boolean queries 9
  1.4 Boolean querying, extend Boolean querying, and ranked retrieval 11
  1.5 References and further reading 13
  1.6 Exercises 14

2 The dictionary and postings lists 17
  2.1 Document delineation and character sequence decoding 17
    2.1.1 Obtaining the character sequence in a document 17
    2.1.2 Choosing a document unit 18
  2.2 Determining dictionary terms 20
    2.2.1 Tokenization 20
    2.2.2 Dropping common terms: stop words 23
    2.2.3 Normalization (equivalence classing of terms) 24
    2.2.4 Stemming and lemmatization 28
  2.3 Postings lists, revisited 31
    2.3.1 Faster postings merges: Skip pointers 31
    2.3.2 Phrase queries 32
  2.4 References and further reading 36
  2.5 Exercises 37
3 Tolerant retrieval 39
  3.1 Wildcard queries 39
    3.1.1 General wildcard queries 40
    3.1.2 k-gram indexes 42
  3.2 Spelling correction 43
    3.2.1 Implementing spelling correction 43
    3.2.2 Forms of spell correction 44
    3.2.3 Edit distance 44
    3.2.4 k-gram indexes 46
    3.2.5 Context sensitive spelling correction 47
  3.3 Phonetic correction 48
  3.4 References and further reading 50

4 Index construction 51
  4.1 Construction of large indexes 51
  4.2 Distributed indexing 54
  4.3 Dynamic indexing 57
  4.4 Other types of indexes 59
  4.5 References and further reading 60
  4.6 Exercises 61

5 Index compression 65
  5.1 Statistical properties of terms in information retrieval 65
  5.2 Dictionary compression 68
    5.2.1 Dictionary-as-a-string 68
    5.2.2 Blocked storage 70
  5.3 Postings file compression 72
    5.3.1 Variable byte codes 73
    5.3.2 $\gamma$ codes 74
  5.4 References and further reading 80
  5.5 Exercises 81

6 Scoring and term weighting 85
  6.1 Parametric and zone indexes 85
    6.1.1 Weighted zone scoring 87
  6.2 Term frequency and weighting 87
    6.2.1 Inverse document frequency 88
    6.2.2 tf-idf weighting 90
  6.3 Variants in weighting functions 90
    6.3.1 Sublinear tf scaling 91
    6.3.2 Maximum tf normalization 91
    6.3.3 The effect of document length 92
    6.3.4 Learning weight functions 92
Contents

6.3.5 Query-term proximity 94

7 Vector space retrieval 97
7.1 Documents as vectors 97
7.1.1 Inner products 97
7.1.2 Queries as vectors 100
7.1.3 Pivoted normalized document length 101
7.2 Heuristics for efficient scoring and ranking 103
7.2.1 Inexact top K document retrieval 104
7.3 Interaction between vector space and other retrieval methods 107
7.3.1 Query parsing and composite scoring 109
7.4 References and further reading 110

8 Evaluation in information retrieval 111
8.1 Evaluating information retrieval systems and search engines 112
8.2 Standard test collections 113
8.3 Evaluation of unranked retrieval sets 114
8.4 Evaluation of ranked retrieval results 117
8.5 Assessing relevance 121
8.5.1 Document relevance: critiques and justifications of the concept 122
8.5.2 Evaluation heuristics 124
8.6 A broader perspective: System quality and user utility 124
8.6.1 System issues 125
8.6.2 User utility 125
8.7 Results snippets 126
8.8 References and further reading 129

9 Relevance feedback and query expansion 133
9.1 Relevance feedback and pseudo-relevance feedback 134
9.1.1 The Rocchio Algorithm 136
9.1.2 Probabilistic relevance feedback 140
9.1.3 When does relevance feedback work? 141
9.1.4 Relevance Feedback on the Web 142
9.1.5 Evaluation of relevance feedback strategies 143
9.1.6 Pseudo-relevance feedback 143
9.1.7 Indirect relevance feedback 144
9.1.8 Summary 144
9.2 Global methods for query reformulation 145
9.2.1 Vocabulary tools for query reformulation 145
9.2.2 Query expansion 145
Contents

13.5.2 $\chi^2$ feature selection 206
13.5.3 Frequency-based feature selection 208
13.5.4 Comparison of feature selection methods 209
13.6 Evaluation of text classification 210
13.7 References and further reading 212
13.8 Exercises 213

14 Vector space classification 217
14.1 Rocchio classification 218
14.2 $k$ nearest neighbor 223
14.3 Linear vs. nonlinear classifiers and the bias-variance tradeoff 226
  14.3.1 More than two classes 231
14.4 References and further reading 234
14.5 Exercises 234

15 Support vector machines and kernel functions 237
15.1 Support vector machines: The linearly separable case 237
15.2 Soft margin classification 243
15.3 Nonlinear SVMs 244
15.4 Experimental data 247
15.5 Issues in the categorization of text documents 249
15.6 References and further reading 249

16 Flat clustering 251
16.1 Clustering in information retrieval 252
16.2 Problem statement 256
16.3 Evaluation of clustering 257
16.4 K-means 260
  16.4.1 Cluster cardinality in k-means 264
16.5 Model-based clustering 266
16.6 References and further reading 270
16.7 Exercises 271

17 Hierarchical clustering 275
17.1 Hierarchical agglomerative clustering 276
17.2 Single-link and complete-link clustering 279
  17.2.1 Time complexity 283
17.3 Group-average agglomerative clustering 286
17.4 Centroid clustering 287
17.5 Cluster labeling 289
17.6 Variants 291
17.7 Implementation notes 292
17.8 References and further reading 293
17.9 Exercises 294

18 Dimensionality reduction and Latent Semantic Indexing 297
18.1 Linear algebra review 297
18.1.1 Matrix decompositions 300
18.2 Term-document matrices and singular value decompositions 301
18.3 Low rank approximations and latent semantic indexing 302
18.4 References and further reading 305

19 Web search basics 307
19.1 Background and history 307
19.2 Web characteristics 309
19.2.1 Spam 311
19.3 Advertising as the economic model 313
19.4 The search user experience 315
19.4.1 User query needs 315
19.5 Index size and estimation 316
19.6 Duplication and mirrors 319
19.6.1 Shingling 320
19.7 References and further reading 321

20 Web crawling and indexes 323
20.1 Overview 323
20.1.1 Features a crawler must provide 323
20.1.2 Features a crawler should provide 324
20.2 Crawling 324
20.2.1 Crawler architecture 325
20.2.2 DNS resolution 328
20.2.3 The URL frontier 329
20.3 Distributing indexes 332
20.4 Connectivity servers 333
20.5 References and further reading 336

21 Link analysis 337
21.1 The web as a graph 337
21.1.1 Anchor text 338
21.2 Pagerank 340
21.2.1 Markov chain review 341
21.2.2 The Pagerank computation 344
21.2.3 Topic-specific Pagerank 347
21.3 Hubs and Authorities 349
21.3.1 Choosing the subset of the web 352
21.4 References and further reading 353
List of Tables

4.1 Collection statistics for Reuters-RCV1. 52
4.2 System parameters for the exercises. 61
4.3 The five steps in constructing an index for Reuters-RCV1 in block merge indexing. 62
4.4 Collection statistics for a large collection. 62

5.1 The effect of preprocessing on the size of the inverted index for Reuters-RCV1. 66
5.2 Dictionary compression for Reuters-RCV1. 72
5.3 Encoding gaps instead of document ids. 73
5.4 Variable byte (VB) encoding. 73
5.5 Some examples of unary and $\gamma$ codes. 75
5.6 Index and dictionary compression for Reuters-RCV1. 79
5.7 Two gap sequences to be merged in block merge indexing. 83

8.1 Calculation of 11-point Interpolated Average Precision. 118
8.2 Calculating the kappa statistic. 121

10.1 Databases, information retrieval and semistructured retrieval. 152
10.2 INEX 2002 collection statistics. 161
10.3 Selected INEX results for content-and-structure (CAS) queries and the quantization function $q$. 164

13.1 Training and test times for Naive Bayes. 201
13.2 Multinomial vs. Binomial model. 202
13.3 Correct estimation implies accurate prediction, but accurate prediction does not imply correct estimation. 203
13.4 Critical values of the $\chi^2$ distribution with one degree of freedom. 207
<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.5</td>
<td>The ten largest classes in the Reuters-21578 collection with number of documents in training and test sets.</td>
<td>210</td>
</tr>
<tr>
<td>13.6</td>
<td>Macro- and microaveraging.</td>
<td>211</td>
</tr>
<tr>
<td>13.7</td>
<td>Experimental results for $F_1$ on Reuters-21578 (all classes).</td>
<td>212</td>
</tr>
<tr>
<td>13.8</td>
<td>A set of documents for which the Naive Bayes independence assumptions are problematic.</td>
<td>213</td>
</tr>
<tr>
<td>14.1</td>
<td>Training and test time for Rocchio classification.</td>
<td>222</td>
</tr>
<tr>
<td>14.2</td>
<td>Training and test time for kNN classification.</td>
<td>224</td>
</tr>
<tr>
<td>14.3</td>
<td>A linear classifier.</td>
<td>227</td>
</tr>
<tr>
<td>14.4</td>
<td>A confusion matrix for Reuters-21578.</td>
<td>233</td>
</tr>
<tr>
<td>15.1</td>
<td>SVM classifier break-even performance from Dumais et al. (1998).</td>
<td>248</td>
</tr>
<tr>
<td>15.2</td>
<td>SVM classifier break-even performance from Joachims (1998).</td>
<td>248</td>
</tr>
<tr>
<td>16.1</td>
<td>Some applications of clustering in information retrieval.</td>
<td>253</td>
</tr>
<tr>
<td>16.2</td>
<td>The EM clustering algorithm.</td>
<td>269</td>
</tr>
<tr>
<td>17.1</td>
<td>Comparison of HAC algorithms.</td>
<td>279</td>
</tr>
<tr>
<td>17.2</td>
<td>Automatically computed cluster labels for a k-means clustering ($K = 10$) of the first 10,000 documents in Reuters-RCV.</td>
<td>290</td>
</tr>
</tbody>
</table>
List of Figures

1.1 A term-document incidence matrix. 3
1.2 Results from Shakespeare for the query Brutus AND Caesar AND NOT Calpurnia. 4
1.3 The two parts of an inverted index. 5
1.4 Indexing, part one. 7
1.5 Indexing, part two. 8
1.6 Merging the postings lists for Brutus and Calpurnia from Figure 1.3. 9
1.7 Algorithm for the merging (or intersection) of two postings lists. 10
1.8 Algorithm for merging (or intersection) of a set of postings lists. 11
1.9 Example Boolean queries on WestLaw. 13

2.1 Arabic script example. 19
2.2 The conceptual linear order of characters is not necessarily the order that you see on the page. 19
2.3 Chinese text example. 23
2.4 A stop list of 25 words common in Reuters-RCV1. 23
2.5 An example of how asymmetric expansion of query terms can usefully model users’ expectations. 25
2.6 Example of Japanese text. 27
2.7 A comparison of three stemming algorithms on a sample text. 30
2.8 Postings lists with skip pointers. 31
2.9 Positional index example. 34

3.1 Example of an entry in the permuterm index. 41
3.2 Example of a postings list in a 3-gram index. 42
3.3 Dynamic programming algorithm for computing the edit distance between strings s1 and s2. 45
3.4 Matching at least two of the three 2-grams in the query bord. 46
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Document from the Reuters newswire.</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Merging in block merge indexing.</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>An example of distributed indexing with MapReduce.</td>
<td>55</td>
</tr>
<tr>
<td>4.4</td>
<td>Map and reduce functions in MapReduce.</td>
<td>57</td>
</tr>
<tr>
<td>4.5</td>
<td>Logarithmic merging.</td>
<td>58</td>
</tr>
<tr>
<td>4.6</td>
<td>Access control lists in retrieval.</td>
<td>60</td>
</tr>
<tr>
<td>5.1</td>
<td>Heaps’ law.</td>
<td>67</td>
</tr>
<tr>
<td>5.2</td>
<td>Storing the dictionary as an array of fixed-width entries.</td>
<td>68</td>
</tr>
<tr>
<td>5.3</td>
<td>Dictionary-as-a-string storage.</td>
<td>69</td>
</tr>
<tr>
<td>5.4</td>
<td>Blocked storage with four terms per block.</td>
<td>70</td>
</tr>
<tr>
<td>5.5</td>
<td>Search of the uncompressed dictionary.</td>
<td>71</td>
</tr>
<tr>
<td>5.6</td>
<td>Search of a dictionary compressed by blocking with $k = 4$.</td>
<td>71</td>
</tr>
<tr>
<td>5.7</td>
<td>Front coding.</td>
<td>71</td>
</tr>
<tr>
<td>5.8</td>
<td>Zipf’s law for Reuters-RCV1.</td>
<td>76</td>
</tr>
<tr>
<td>5.9</td>
<td>Stratification of terms for estimating the size of a $\gamma$ encoded inverted index.</td>
<td>78</td>
</tr>
<tr>
<td>6.1</td>
<td>Basic zone index</td>
<td>86</td>
</tr>
<tr>
<td>6.2</td>
<td>Zone index in which the zone is encoded in the postings rather than the dictionary.</td>
<td>87</td>
</tr>
<tr>
<td>6.3</td>
<td>Collection frequency (cf) and document frequency (df) behave differently.</td>
<td>89</td>
</tr>
<tr>
<td>6.4</td>
<td>Example of idf values.</td>
<td>89</td>
</tr>
<tr>
<td>7.1</td>
<td>Cosine similarity illustrated.</td>
<td>98</td>
</tr>
<tr>
<td>7.2</td>
<td>Term frequencies in three novels.</td>
<td>99</td>
</tr>
<tr>
<td>7.3</td>
<td>Term vectors for the three novels of Figure 7.2.</td>
<td>99</td>
</tr>
<tr>
<td>7.4</td>
<td>Pivoted document length normalization - figure to be completed.</td>
<td>102</td>
</tr>
<tr>
<td>7.5</td>
<td>Implementing pivoted document length normalization by linear scaling.</td>
<td>103</td>
</tr>
<tr>
<td>8.1</td>
<td>Graph comparing the harmonic mean to other means.</td>
<td>116</td>
</tr>
<tr>
<td>8.2</td>
<td>Precision/Recall graph.</td>
<td>117</td>
</tr>
<tr>
<td>8.3</td>
<td>Averaged 11-Point Precision/Recall graph across 50 queries for a representative TREC system.</td>
<td>119</td>
</tr>
<tr>
<td>8.4</td>
<td>The ROC curve corresponding to the precision-recall curve in Figure 8.2.</td>
<td>120</td>
</tr>
<tr>
<td>8.5</td>
<td>The dynamic snippet (bold italic) generated for a document for the query new guinea economic development, showing where it occurs in the original document.</td>
<td>128</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>9.1</td>
<td>An example of relevance feedback searching over images.</td>
<td>134</td>
</tr>
<tr>
<td>9.2</td>
<td>An example of relevance feedback searching over images (continued).</td>
<td>135</td>
</tr>
<tr>
<td>9.3</td>
<td>An example of relevance feedback searching over images (continued).</td>
<td>135</td>
</tr>
<tr>
<td>9.4</td>
<td>An example of relevance feedback searching over images (continued).</td>
<td>135</td>
</tr>
<tr>
<td>9.5</td>
<td>Example of relevance feedback on a text collection.</td>
<td>136</td>
</tr>
<tr>
<td>9.6</td>
<td>The Rocchio optimal query for separating relevant and non-relevant documents.</td>
<td>137</td>
</tr>
<tr>
<td>9.7</td>
<td>An application of Rocchio’s algorithm.</td>
<td>138</td>
</tr>
<tr>
<td>9.8</td>
<td>Results showing pseudo relevance feedback greatly improving performance.</td>
<td>139</td>
</tr>
<tr>
<td>9.9</td>
<td>An example of query expansion in the interface of the Yahoo! web search engine in 2006.</td>
<td>144</td>
</tr>
<tr>
<td>9.10</td>
<td>Examples of query expansion via the PubMed thesaurus.</td>
<td>146</td>
</tr>
<tr>
<td>9.11</td>
<td>An example of an automatically generated thesaurus.</td>
<td>148</td>
</tr>
<tr>
<td>10.1</td>
<td>An XML document.</td>
<td>152</td>
</tr>
<tr>
<td>10.2</td>
<td>The XML document in Figure 10.1 as a DOM object.</td>
<td>153</td>
</tr>
<tr>
<td>10.3</td>
<td>Indexing units in XML retrieval.</td>
<td>155</td>
</tr>
<tr>
<td>10.4</td>
<td>A schema mismatch.</td>
<td>156</td>
</tr>
<tr>
<td>10.5</td>
<td>XML queries.</td>
<td>157</td>
</tr>
<tr>
<td>10.6</td>
<td>A mapping of an XML document (left) to a set of structural terms (right).</td>
<td>158</td>
</tr>
<tr>
<td>10.7</td>
<td>Query-document matching for extended queries.</td>
<td>159</td>
</tr>
<tr>
<td>10.8</td>
<td>Inverted index search for extended queries.</td>
<td>160</td>
</tr>
<tr>
<td>10.9</td>
<td>Indexing and search in JuruXML.</td>
<td>160</td>
</tr>
<tr>
<td>10.10</td>
<td>Schema of the documents in the INEX collection.</td>
<td>162</td>
</tr>
<tr>
<td>10.11</td>
<td>An INEX CAS topic.</td>
<td>162</td>
</tr>
<tr>
<td>11.1</td>
<td>A tree of dependencies between terms.</td>
<td>176</td>
</tr>
<tr>
<td>12.1</td>
<td>A simple finite automaton and some of the strings in the language that it generates.</td>
<td>182</td>
</tr>
<tr>
<td>12.2</td>
<td>A one-state finite automaton that acts as a unigram language model together with a partial specification of the state emission probabilities.</td>
<td>182</td>
</tr>
<tr>
<td>12.3</td>
<td>Partial specification of two unigram language models.</td>
<td>183</td>
</tr>
<tr>
<td>12.4</td>
<td>Results of a comparison of tf-idf to language modeling (LM) term weighting by Ponte and Croft (1998).</td>
<td>188</td>
</tr>
<tr>
<td>12.5</td>
<td>Three ways of developing the language modeling approach: query likelihood, document likelihood and model comparison.</td>
<td>190</td>
</tr>
</tbody>
</table>
List of Figures

13.1 Training and test data in text classification. CHANGE NOTATION: gamma for $c$, $d$ for $x$ 196
13.2 Naive Bayes conditional independence assumption. 198
13.3 Naive Bayes algorithm (multinomimal model): Training and testing 201
13.4 Features with high mutual information scores for six Reuters-RCV1 classes. 205
13.5 Effect of feature set size on accuracy for multinomial and binomial (multivariate Bernoulli) models. 206
13.6 A sample document from the Reuters-21578 collection. 210
14.1 Vector space classification into three classes. 219
14.2 Rocchio classification. 220
14.3 Rocchio classification: Training and testing. 221
14.4 A polymorphic class with two different centers. 222
14.5 kNN classification. 223
14.6 kNN training and testing. 224
14.7 There is an infinite number of hyperplanes that separate two linearly separable classes. 226
14.8 The bias-variance tradeoff. 230
14.9 Linear classification for one-of classification with $J > 2$. 232
14.10 Example for differences between Euclidean distance, inner product similarity and cosine similarity. 235
14.11 A simple non-separable set of points. 236
15.1 The intuition of large-margin classifiers. 238
15.2 Support vectors are the points right up against the margin of the classifier. 239
15.3 The geometric margin of a linear classifier. 240
15.4 Large margin classification with slack variables. 243
15.5 Projecting nonlinearly separable data into a higher dimensional space can make it linearly separable. 245
16.1 An example of a data set with a clear cluster structure. 251
16.2 Clustering of search results to improve user recall. 254
16.3 The Scatter-Gather user interface. 255
16.4 Purity as an external evaluation criterion for cluster quality. 258
16.5 The k-means algorithm. 261
16.6 One iteration of the k-means algorithm in $\mathbb{R}^2$. 261
16.7 The outcome of clustering in k-means depends on the initial seeds. 263
16.8 Minimal residual sum of squares (RSS) as a function of the number of clusters in k-means. 264
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.1</td>
<td>A dendrogram of a single-link clustering of 30 documents from Reuters-RCV1.</td>
<td>277</td>
</tr>
<tr>
<td>17.2</td>
<td>A simple, but inefficient HAC algorithm.</td>
<td>278</td>
</tr>
<tr>
<td>17.3</td>
<td>A single-link (top) and complete-link (bottom) clustering of eight documents.</td>
<td>280</td>
</tr>
<tr>
<td>17.4</td>
<td>A dendrogram of a complete-link clustering of 30 documents from Reuters-RCV1.</td>
<td>281</td>
</tr>
<tr>
<td>17.5</td>
<td>Chaining in single-link clustering.</td>
<td>282</td>
</tr>
<tr>
<td>17.6</td>
<td>Outliers in complete-link clustering.</td>
<td>283</td>
</tr>
<tr>
<td>17.7</td>
<td>Single-link clustering algorithm using an NBM array.</td>
<td>284</td>
</tr>
<tr>
<td>17.8</td>
<td>A generic HAC algorithm.</td>
<td>285</td>
</tr>
<tr>
<td>17.9</td>
<td>Complete-link clustering is not best-merge persistent.</td>
<td>286</td>
</tr>
<tr>
<td>17.10</td>
<td>Centroid clustering.</td>
<td>288</td>
</tr>
<tr>
<td>17.11</td>
<td>Centroid clustering is not monotonic.</td>
<td>289</td>
</tr>
<tr>
<td>17.12</td>
<td>Single-link clustering of points on a line.</td>
<td>295</td>
</tr>
<tr>
<td>18.1</td>
<td>Original and LSI spaces. Only two of many axes are shown in each case.</td>
<td>305</td>
</tr>
<tr>
<td>20.1</td>
<td>The basic crawler architecture.</td>
<td>326</td>
</tr>
<tr>
<td>20.2</td>
<td>An example robots.txt file.</td>
<td>327</td>
</tr>
<tr>
<td>20.3</td>
<td>Distributing the basic crawl architecture.</td>
<td>328</td>
</tr>
<tr>
<td>20.4</td>
<td>The URL frontier.</td>
<td>330</td>
</tr>
<tr>
<td>20.5</td>
<td>Example of an auxiliary hosts-to-back queues table.</td>
<td>331</td>
</tr>
<tr>
<td>20.6</td>
<td>A segment of the lexicographic ordering of all URLs.</td>
<td>334</td>
</tr>
<tr>
<td>20.7</td>
<td>A four-row segment of the table of links.</td>
<td>335</td>
</tr>
<tr>
<td>21.1</td>
<td>Two nodes of the web graph joined by a link.</td>
<td>338</td>
</tr>
<tr>
<td>21.2</td>
<td>A fragment of html code.</td>
<td>338</td>
</tr>
<tr>
<td>21.3</td>
<td>The random surfer at node A proceeds with probability 1/3 to each of B, C and D.</td>
<td>341</td>
</tr>
<tr>
<td>21.4</td>
<td>A simple Markov chain with three states; the numbers on the links indicate the transition probabilities.</td>
<td>342</td>
</tr>
<tr>
<td>21.5</td>
<td>A small web graph.</td>
<td>346</td>
</tr>
</tbody>
</table>
Acknowledgments 1
Dinquan Chen, Elmer Garduno, Sergio Govoni, Corinna Habets, Ralf Jankowitsch, Vinay Kakade, Mei Kobayashi, Wessel Kraaij, Paul McNamee, Scot Olsson, Klaus RothenhÄusler, Helmut Schmid, Jason Utt, Mike Walsh, Changliang Wang, Thomas Zeume,

Acknowledgments 2
Omar Alonso, Vo Ngoc Anh, Eric Brown, Stefan Bütcher, Doug Cutting, Byron Dom, Johannes Fürnkranz, Gonzalo Navarro, Paul Ogilvie, Jan Pedersen, Sabine Schulte im Walde, John Tait, Justin Zobel,
Table of Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma$</td>
<td>$\gamma$ code</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Classification or clustering function: $\gamma(d)$ is $d$’s class or cluster</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Weight of negative documents in Rocchio relevance feedback</td>
</tr>
<tr>
<td>$\gamma_k$</td>
<td>Prior in EM clustering for cluster $k$</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>Set of classes ${c_1, \ldots, c_J}$</td>
</tr>
<tr>
<td>$\omega_k$</td>
<td>Cluster $\omega_k$ in classification where</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Clustering of set of clusters ${\omega_1, \ldots, \omega_K}$</td>
</tr>
<tr>
<td>$\arg\max_x f(x)$</td>
<td>The value of $x$ for which $f$ reaches its maximum</td>
</tr>
<tr>
<td>$\arg\min_x f(x)$</td>
<td>The value of $x$ for which $f$ reaches its minimum</td>
</tr>
<tr>
<td>$c, c_j$</td>
<td>Class or category in classification</td>
</tr>
<tr>
<td>$c(w)$</td>
<td>Count of word $w$ (Chapter 12)</td>
</tr>
<tr>
<td>$C$</td>
<td>Set ${c_1, \ldots, c_j, \ldots, c_J}$ of all classes</td>
</tr>
<tr>
<td>$d$</td>
<td>Document</td>
</tr>
<tr>
<td>$</td>
<td>d</td>
</tr>
<tr>
<td>$\vec{d}$</td>
<td>Document vector</td>
</tr>
<tr>
<td>$</td>
<td>\vec{d}</td>
</tr>
<tr>
<td>$D$</td>
<td>Set ${d_1, \ldots, d_n, \ldots, d_N}$ of all documents (or document vectors)</td>
</tr>
<tr>
<td>$f_i$</td>
<td>Frequency of term with rank $i$</td>
</tr>
<tr>
<td>$H$</td>
<td>Entropy; harmonic number</td>
</tr>
<tr>
<td>$J$</td>
<td>Number of classes</td>
</tr>
<tr>
<td>$k$</td>
<td>Top $k$ retrieved documents; top $k$ selected features from the vocabulary $V$</td>
</tr>
<tr>
<td>$k$</td>
<td>Number of nearest neighbors in kNN</td>
</tr>
</tbody>
</table>
Table of Notations

$K$  
Number of clusters

$L, L_d, L_t$  
Average length of a document, training document, and test document (in tokens)

$M$  
Size of vocabulary ($|V|$)

$N$  
Number of documents in the retrieval collection

$N_j$  
Number of documents in class $c_j$

$N(\omega)$  
Number of times the event $\omega$ occurred

$n$  
Number of attributes in the representation of $d$: $<x_1, x_2, \ldots, x_n>$

$n$  
Number of postings

$S$  
Positions in the test document that contain words from the vocabulary $V$

$T$  
Total number of tokens in the document collection

$T_j$  
Number of tokens in documents in class $c_j$

$T_{ji}$  
Number of occurrences of word $i$ in class $c_j$

$t$  
Structural term (word + context in XML retrieval)

$\vec{u}_j$  
Rocchio centroid of class $c_j$

$V$  
Vocabulary $\{w_1, \ldots, w_i, \ldots, w_M\}$ of terms indexed

$\vec{V}(d)$  
Unnormalized document vector

$\vec{v}(d)$  
Normalized document vector

$\vec{w}^T \vec{x} = b$  
Hyperplane

$\vec{x} = (x_1, \ldots, x_i, \ldots, x_M)$  
Term incidence vector; more generally: document feature representation

$X_i$  
Random variable for attribute $i$

$X$  
Instance space in text classification

$x_i$  
Sequence $\{x_1, \ldots, x_i, \ldots, x_{L_t}\}$ of tokens in the test document

$|\vec{x} - \vec{y}|$  
Euclidean distance of $\vec{x}$ and $\vec{y}$

$\vec{V}(d)$  
Vector of document $d$
Preface

As recently as the 1990s, studies showed that most people preferred getting information from other people rather than information retrieval systems. Of course, in that time period, most people also interacted with human travel agents to book their travel. While academic discussion of this process is unfortunately scant, in the last decade, relentless optimization of formal measures of performance has driven web search engines to new levels of performance where most people are satisfied most of the time, and web search has become a standard and often preferred source of information finding. For example, the 2004 Pew Internet Survey (Fallows 2004) found that “92% of Internet users say the Internet is a good place to go for getting everyday information.” To the surprise of many, the field of information retrieval has moved from being a sleepy backwater to being most people’s preferred means of information access.