

An  
Introduction  
to  
Information  
Retrieval

Draft of April 1, 2009

Online edition (c) 2009 Cambridge UP



An  
Introduction  
to  
Information  
Retrieval

Christopher D. Manning  
Prabhakar Raghavan  
Hinrich Schütze

Cambridge University Press  
Cambridge, England

Online edition (c) 2009 Cambridge UP

DRAFT!

DO NOT DISTRIBUTE WITHOUT PRIOR PERMISSION

© 2009 Cambridge University Press

By Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze

Printed on April 1, 2009

Website: <http://www.informationretrieval.org/>

Comments, corrections, and other feedback most welcome at:

[informationretrieval@yahoogroups.com](mailto:informationretrieval@yahoogroups.com)

Online edition (c) 2009 Cambridge UP

## *Brief Contents*

1	<i>Boolean retrieval</i>	1
2	<i>The term vocabulary and postings lists</i>	19
3	<i>Dictionaries and tolerant retrieval</i>	49
4	<i>Index construction</i>	67
5	<i>Index compression</i>	85
6	<i>Scoring, term weighting and the vector space model</i>	109
7	<i>Computing scores in a complete search system</i>	135
8	<i>Evaluation in information retrieval</i>	151
9	<i>Relevance feedback and query expansion</i>	177
10	<i>XML retrieval</i>	195
11	<i>Probabilistic information retrieval</i>	219
12	<i>Language models for information retrieval</i>	237
13	<i>Text classification and Naive Bayes</i>	253
14	<i>Vector space classification</i>	289
15	<i>Support vector machines and machine learning on documents</i>	319
16	<i>Flat clustering</i>	349
17	<i>Hierarchical clustering</i>	377
18	<i>Matrix decompositions and latent semantic indexing</i>	403
19	<i>Web search basics</i>	421
20	<i>Web crawling and indexes</i>	443
21	<i>Link analysis</i>	461



## Contents

<i>Table of Notation</i>	xv
<i>Preface</i>	xix
<b>1 Boolean retrieval</b>	<b>1</b>
1.1 An example information retrieval problem	3
1.2 A first take at building an inverted index	6
1.3 Processing Boolean queries	10
1.4 The extended Boolean model versus ranked retrieval	14
1.5 References and further reading	17
<b>2 The term vocabulary and postings lists</b>	<b>19</b>
2.1 Document delineation and character sequence decoding	19
2.1.1 Obtaining the character sequence in a document	19
2.1.2 Choosing a document unit	20
2.2 Determining the vocabulary of terms	22
2.2.1 Tokenization	22
2.2.2 Dropping common terms: stop words	27
2.2.3 Normalization (equivalence classing of terms)	28
2.2.4 Stemming and lemmatization	32
2.3 Faster postings list intersection via skip pointers	36
2.4 Positional postings and phrase queries	39
2.4.1 Biword indexes	39
2.4.2 Positional indexes	41
2.4.3 Combination schemes	43
2.5 References and further reading	45
<b>3 Dictionaries and tolerant retrieval</b>	<b>49</b>
3.1 Search structures for dictionaries	49
3.2 Wildcard queries	51
3.2.1 General wildcard queries	53

3.2.2	<i>k</i> -gram indexes for wildcard queries	54
3.3	Spelling correction	56
3.3.1	Implementing spelling correction	57
3.3.2	Forms of spelling correction	57
3.3.3	Edit distance	58
3.3.4	<i>k</i> -gram indexes for spelling correction	60
3.3.5	Context sensitive spelling correction	62
3.4	Phonetic correction	63
3.5	References and further reading	65
4	<i>Index construction</i>	67
4.1	Hardware basics	68
4.2	Blocked sort-based indexing	69
4.3	Single-pass in-memory indexing	73
4.4	Distributed indexing	74
4.5	Dynamic indexing	78
4.6	Other types of indexes	80
4.7	References and further reading	83
5	<i>Index compression</i>	85
5.1	Statistical properties of terms in information retrieval	86
5.1.1	Heaps' law: Estimating the number of terms	88
5.1.2	Zipf's law: Modeling the distribution of terms	89
5.2	Dictionary compression	90
5.2.1	Dictionary as a string	91
5.2.2	Blocked storage	92
5.3	Postings file compression	95
5.3.1	Variable byte codes	96
5.3.2	$\gamma$ codes	98
5.4	References and further reading	105
6	<i>Scoring, term weighting and the vector space model</i>	109
6.1	Parametric and zone indexes	110
6.1.1	Weighted zone scoring	112
6.1.2	Learning weights	113
6.1.3	The optimal weight <i>g</i>	115
6.2	Term frequency and weighting	117
6.2.1	Inverse document frequency	117
6.2.2	Tf-idf weighting	118
6.3	The vector space model for scoring	120
6.3.1	Dot products	120
6.3.2	Queries as vectors	123
6.3.3	Computing vector scores	124



6.4	Variant tf-idf functions	126
6.4.1	Sublinear tf scaling	126
6.4.2	Maximum tf normalization	127
6.4.3	Document and query weighting schemes	128
6.4.4	Pivoted normalized document length	129
6.5	References and further reading	133
<b>7</b>	<b><i>Computing scores in a complete search system</i></b>	<b>135</b>
7.1	Efficient scoring and ranking	135
7.1.1	Inexact top <i>K</i> document retrieval	137
7.1.2	Index elimination	137
7.1.3	Champion lists	138
7.1.4	Static quality scores and ordering	138
7.1.5	Impact ordering	140
7.1.6	Cluster pruning	141
7.2	Components of an information retrieval system	143
7.2.1	Tiered indexes	143
7.2.2	Query-term proximity	144
7.2.3	Designing parsing and scoring functions	145
7.2.4	Putting it all together	146
7.3	Vector space scoring and query operator interaction	147
7.4	References and further reading	149
<b>8</b>	<b><i>Evaluation in information retrieval</i></b>	<b>151</b>
8.1	Information retrieval system evaluation	152
8.2	Standard test collections	153
8.3	Evaluation of unranked retrieval sets	154
8.4	Evaluation of ranked retrieval results	158
8.5	Assessing relevance	164
8.5.1	Critiques and justifications of the concept of relevance	166
8.6	A broader perspective: System quality and user utility	168
8.6.1	System issues	168
8.6.2	User utility	169
8.6.3	Refining a deployed system	170
8.7	Results snippets	170
8.8	References and further reading	173
<b>9</b>	<b><i>Relevance feedback and query expansion</i></b>	<b>177</b>
9.1	Relevance feedback and pseudo relevance feedback	178
9.1.1	The Rocchio algorithm for relevance feedback	178
9.1.2	Probabilistic relevance feedback	183
9.1.3	When does relevance feedback work?	183

9.1.4	Relevance feedback on the web	185	
9.1.5	Evaluation of relevance feedback strategies	186	
9.1.6	Pseudo relevance feedback	187	
9.1.7	Indirect relevance feedback	187	
9.1.8	Summary	188	
9.2	Global methods for query reformulation	189	
9.2.1	Vocabulary tools for query reformulation	189	
9.2.2	Query expansion	189	
9.2.3	Automatic thesaurus generation	192	
9.3	References and further reading	193	
<b>10</b>	<b><i>XML retrieval</i></b>	<b>195</b>	
10.1	Basic XML concepts	197	
10.2	Challenges in XML retrieval	201	
10.3	A vector space model for XML retrieval	206	
10.4	Evaluation of XML retrieval	210	
10.5	Text-centric vs. data-centric XML retrieval	214	
10.6	References and further reading	216	
10.7	Exercises	217	
<b>11</b>	<b><i>Probabilistic information retrieval</i></b>	<b>219</b>	
11.1	Review of basic probability theory	220	
11.2	The Probability Ranking Principle	221	
11.2.1	The 1/0 loss case	221	
11.2.2	The PRP with retrieval costs	222	
11.3	The Binary Independence Model	222	
11.3.1	Deriving a ranking function for query terms	224	
11.3.2	Probability estimates in theory	226	
11.3.3	Probability estimates in practice	227	
11.3.4	Probabilistic approaches to relevance feedback	228	
11.4	An appraisal and some extensions	230	
11.4.1	An appraisal of probabilistic models	230	
11.4.2	Tree-structured dependencies between terms	231	
11.4.3	Okapi BM25: a non-binary model	232	
11.4.4	Bayesian network approaches to IR	234	
11.5	References and further reading	235	
<b>12</b>	<b><i>Language models for information retrieval</i></b>	<b>237</b>	
12.1	Language models	237	
12.1.1	Finite automata and language models	237	
12.1.2	Types of language models	240	
12.1.3	Multinomial distributions over words	241	
12.2	The query likelihood model	242	