

# Modern Information Retrieval

**Ricardo Baeza-Yates**  
**Berthier Ribeiro-Neto**



ACM Press  
New York



Addison-Wesley

Harlow, England • Reading, Massachusetts  
Menlo Park, California • New York  
Don Mills, Ontario • Amsterdam • Bonn  
Sydney • Singapore • Tokyo • Madrid  
San Juan • Milan • Mexico City • Seoul • Taipei

Copyright © 1999 by the ACM press, A Division of the Association for Computing Machinery, Inc. (ACM).

Addison Wesley Longman Limited  
Edinburgh Gate  
Harlow  
Essex CM20 2JE  
England

and Associated Companies throughout the World.

The rights of the authors of this Work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without either the prior written permission of the publisher or a licence permitting restricted copying in the United Kingdom issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London W1P 9HE.

While the publisher has made every attempt to trace all copyright owners and obtain permission to reproduce material, in a few cases this has proved impossible. Copyright holders of material which has not been acknowledged are encouraged to contact the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Addison Wesley Longman Limited has made every attempt to supply trade mark information about manufacturers and their products mentioned in this book. A list of the trademark designations and their owners appears on page viii.

Typeset in Computer Modern by 56  
Printed and bound in the United States of America

First printed 1999

ISBN 0-201-39829-X

**British Library Cataloguing-in-Publication Data**

A catalogue record for this book is available from the British Library

**Library of Congress Cataloguing-in-Publication Data**

Baeza-Yates, R.(Ricardo)

Modern information retrieval / Ricardo Baeza-Yates, Berthier Ribeiro-Neto.

p. cm.

Includes bibliographical references and index.

ISBN 0-201-39829-X

1. Information storage and retrieval systems. I. Ribeiro, Berthier de Araújo Neto, 1960- . II. Title.

Z667.B34 1999

025.04-dc21

99-10033

CIP

$t$ -dimensional vectorial space and standard linear algebra operations on vectors. For the classic probabilistic model, the framework is composed of sets, standard probability operations, and the Bayes' theorem.

In the remainder of this chapter, we discuss the various IR models shown in Figure 2.1. Throughout the discussion, we do not explicitly instantiate the components  $\mathbf{D}$ ,  $\mathbf{Q}$ ,  $\mathcal{F}$ , and  $R(q_i, d_j)$  of each model. Such components should be quite clear from the discussion and can be easily inferred.

## 2.5 Classic Information Retrieval

In this section we briefly present the three classic models in information retrieval namely, the Boolean, the vector, and the probabilistic models.

### 2.5.1 Basic Concepts

The classic models in information retrieval consider that each document is described by a set of representative keywords called index terms. An *index term* is simply a (document) word whose semantics helps in remembering the document's main themes. Thus, index terms are used to index and summarize the document contents. In general, index terms are mainly nouns because nouns have meaning by themselves and thus, their semantics is easier to identify and to grasp. Adjectives, adverbs, and connectives are less useful as index terms because they work mainly as complements. However, it might be interesting to consider all the distinct words in a document collection as index terms. For instance, this approach is adopted by some Web search engines as discussed in Chapter 13 (in which case, the document logical view is *full text*). We postpone a discussion on the problem of how to generate index terms until Chapter 7, where the issue is covered in detail.

Given a set of index terms for a document, we notice that not all terms are equally useful for describing the document contents. In fact, there are index terms which are simply vaguer than others. Deciding on the importance of a term for summarizing the contents of a document is not a trivial issue. Despite this difficulty, there are properties of an index term which are easily measured and which are useful for evaluating the potential of a term as such. For instance, consider a collection with a hundred thousand documents. A word which appears in each of the one hundred thousand documents is completely useless as an index term because it does not tell us anything about which documents the user might be interested in. On the other hand, a word which appears in just five documents is quite useful because it narrows down considerably the space of documents which might be of interest to the user. Thus, it should be clear that distinct index terms have varying relevance when used to describe document contents. This effect is captured through the assignment of numerical *weights* to each index term of a document.

Let  $k_i$  be an index term,  $d_j$  be a document, and  $w_{i,j} \geq 0$  be a *weight* associated with the pair  $(k_i, d_j)$ . This weight quantifies the importance of the index term for describing the document semantic contents.

**Definition** Let  $t$  be the number of index terms in the system and  $k_i$  be a generic index term.  $K = \{k_1, \dots, k_t\}$  is the set of all index terms. A weight  $w_{i,j} > 0$  is associated with each index term  $k_i$  of a document  $d_j$ . For an index term which does not appear in the document text,  $w_{i,j} = 0$ . With the document  $d_j$  is associated an index term vector  $\vec{d}_j$  represented by  $\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$ . Further, let  $g_i$  be a function that returns the weight associated with the index term  $k_i$  in any  $t$ -dimensional vector (i.e.,  $g_i(\vec{d}_j) = w_{i,j}$ ).

As we later discuss, the index term weights are usually assumed to be mutually independent. This means that knowing the weight  $w_{i,j}$  associated with the pair  $(k_i, d_j)$  tells us nothing about the weight  $w_{i+1,j}$  associated with the pair  $(k_{i+1}, d_j)$ . This is clearly a simplification because occurrences of index terms in a document are not uncorrelated. Consider, for instance, that the terms *computer* and *network* are used to index a given document which covers the area of computer networks. Frequently, in this document, the appearance of one of these two words attracts the appearance of the other. Thus, these two words are correlated and their weights could reflect this correlation. While mutual independence seems to be a strong simplification, it does simplify the task of computing index term weights and allows for fast ranking computation. Furthermore, taking advantage of index term correlations for improving the final document ranking is not a simple task. In fact, none of the many approaches proposed in the past has clearly demonstrated that index term correlations are advantageous (for ranking purposes) with general collections. Therefore, unless clearly stated otherwise, we assume mutual independence among index terms. In Chapter 5 we discuss modern retrieval techniques which are based on term correlations and which have been tested successfully with particular collections. These successes seem to be slowly shifting the current understanding towards a more favorable view of the usefulness of term correlations for information retrieval systems.

The above definitions provide support for discussing the three classic information retrieval models, namely, the Boolean, the vector, and the probabilistic models, as we now do.

### 2.5.2 Boolean Model

The Boolean model is a simple retrieval model based on set theory and Boolean algebra. Since the concept of a set is quite intuitive, the Boolean model provides a framework which is easy to grasp by a common user of an IR system. Furthermore, the queries are specified as Boolean expressions which have precise semantics. Given its inherent simplicity and neat formalism, the Boolean model received great attention in past years and was adopted by many of the early commercial bibliographic systems.

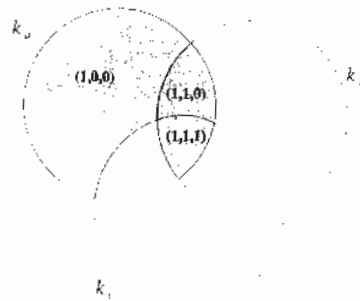


Figure 2.3 The three conjunctive components for the query  $[q = k_a \wedge (k_b \vee \neg k_c)]$ .

Unfortunately, the Boolean model suffers from major drawbacks. First, its retrieval strategy is based on a binary decision criterion (i.e., a document is predicted to be either relevant or non-relevant) without any notion of a grading scale, which prevents good retrieval performance. Thus, the Boolean model is in reality much more a data (instead of information) retrieval model. Second, while Boolean expressions have precise semantics, frequently it is not simple to translate an information need into a Boolean expression. In fact, most users find it difficult and awkward to express their query requests in terms of Boolean expressions. The Boolean expressions actually formulated by users often are quite simple (see Chapter 10 for a more thorough discussion on this issue). Despite these drawbacks, the Boolean model is still the dominant model with commercial document database systems and provides a good starting point for those new to the field.

The Boolean model considers that index terms are present or absent in a document. As a result, the index term weights are assumed to be all binary, i.e.,  $w_{i,j} \in \{0, 1\}$ . A query  $q$  is composed of index terms linked by three connectives: *not*, *and*, or. Thus, a query is essentially a conventional Boolean expression which can be represented as a disjunction of conjunctive vectors (i.e., in *disjunctive normal form* – DNF). For instance, the query  $[q = k_a \wedge (k_b \vee \neg k_c)]$  can be written in disjunctive normal form as  $[\vec{q}_{dnf} = (1, 1, 1) \vee (1, 1, 0) \vee (1, 0, 0)]$ , where each of the components is a binary weighted vector associated with the tuple  $(k_a, k_b, k_c)$ . These binary weighted vectors are called the conjunctive components of  $\vec{q}_{dnf}$ . Figure 2.3 illustrates the three conjunctive components for the query  $q$ .

**Definition** For the Boolean model, the index term weight variables are all binary i.e.,  $w_{i,j} \in \{0, 1\}$ . A query  $q$  is a conventional Boolean expression. Let  $\vec{q}_{dnf}$  be the disjunctive normal form for the query  $q$ . Further, let  $\vec{q}_{cc}$  be any of the conjunctive components of  $\vec{q}_{dnf}$ . The similarity of a document  $d_j$  to the query  $q$  is defined as

$$\text{sim}(d_j, q) = \begin{cases} 1 & \text{if } \exists \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i, g_i(\vec{d}_j) = g_i(\vec{q}_{cc})) \\ 0 & \text{otherwise} \end{cases}$$

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

## LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

## FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

## E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.