

PREFACE

This book describes how an information retrieval system really works. We assume that anyone who has used a web search engine already knows how to *use* a retrieval system – to our knowledge, there is no book about *how* a publicly available search engine works. The book is intended for students in an information retrieval course, but this book may well be of interest to practitioners working on commercial information retrieval systems (as it illuminates the essence of how they work) as well as people who are faced with selecting an information retrieval system. By knowing how these systems work, it is much easier to identify differences between competing products.

There are many different algorithms for finding relevant documents. We describe the most popular algorithms focusing not only on the most efficient in terms of run time as commonly done in the realm of on-line search engines, but also on the most accurate in terms of finding highly relevant documents. We illustrate each feature via the use of a sample system that we have written especially for this book called SimpleIR. This is a Java-based system, and we will describe it in detail. We expect that student projects will involve making modifications to SimpleIR. Hence, by the end of a semester course in information retrieval, students will have actually worked with an information retrieval system and will have an understanding of the tradeoffs involved in trying to improve one.

We expect this book to be used in a senior level undergraduate Information Retrieval course or an early graduate-level course. For a graduate level course, we would expect that the course would be augmented with recent papers in the field and students would be asked to implement new algorithms on top of SimpleIR.

CHAPTER 1: Introduction

With the advent of web search engines and the proliferation of text on the web, the problem of finding relevant documents has become much more visible. Researchers have been working on document retrieval for many years (since the 1950's), but until the advent of the web, this area of research has, for the most part, remained in the shadows of conventional processing. In 1993, an international text search evaluation forum drew about 20-30 participants, whereas in 2000, the same forum had nearly one hundred participants.

Every large company has numerous internal documents in electronic form. The widespread use of scanners and readily available OCR (optical character recognition) software has increased the sea of information both inside and outside of a company. The ability to find some particularly relevant document is no longer just “nice-to-have” – for many organizations it has become crucial to their business.

Web search technology is a specialty within the general field of information retrieval. Information retrieval is the *retrieval* of documents *relevant to user requests*, commonly called *queries*. Information retrieval research and development efforts focus on both *efficient* and *accurate* search techniques, and our focus in this textbook is the clarification and illustration of the techniques used to support such search capability.

Given the rapid increase in the volume of digital text, companies which, until recently, had only a rudimentary interest in information retrieval systems now either have or plan to build *information portals*, namely entries to their organizational systems. Typically, each portal invariably has an information retrieval system. It is currently projected that every Fortune 500 Company will have a portal in place by the year 2003.

The market for this business is estimated at over 2 billion dollars [xxx]. Hence, individuals working in computer science related jobs, it is highly likely they either have or will soon encounter an information retrieval system in their workplace.

Information Retrieval differs from the Rest of Computer Science

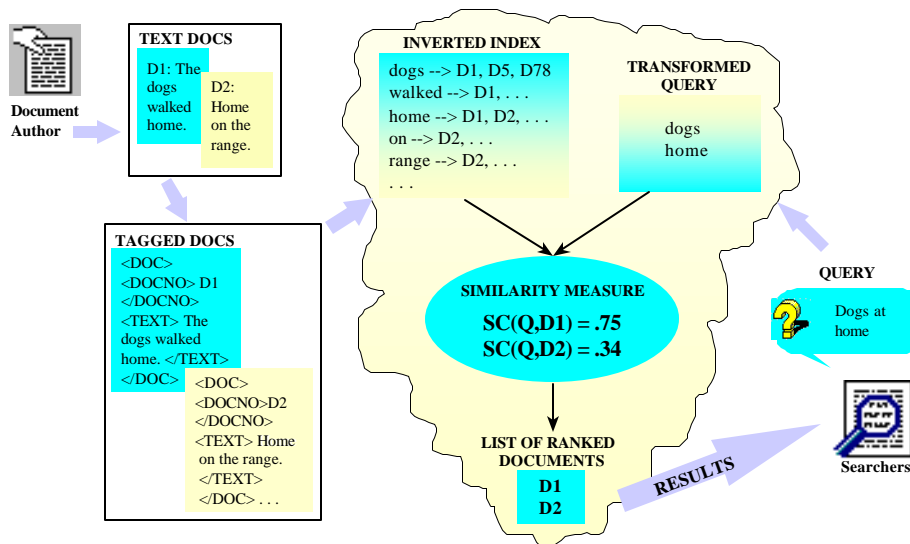
The field of computer science is, for the most part, all about efficiency. From the onset of their studies, students learn simple things like a linear search and they are then immediately taught about the virtues of more efficient searches like a binary search. Moving on to data structures, various structures are taught with the goal of showing students how they can continue to improve efficiency. Finally, in an algorithms course, students are taught how to carefully compute efficiency analyses. For example, various sorting algorithms are deemed “good” or “bad” simply because of efficiency. When we teach information retrieval we find students come into the course intent upon learning the very fastest means of retrieving documents. We have an answer for them – the fastest way to retrieve documents is to retrieve no documents at all. This takes no time, uses no CPU processor, uses no memory, and requires no disk I/O. By definition, this is a correct information retrieval algorithm. The definition of the information retrieval search problem is: Given a query Q , find all the documents in a collection C that are relevant to the query. Hence, retrieving no documents may well be an honest attempt at resolving the query!!

Obviously this is ridiculous, but hopefully you have seen that there is more to information retrieval than simply finding documents quickly – the goal is really to find *relevant* documents *quickly*. This is not simply a matter of doing a pattern-matching request of the terms in the query against the collection. It turns out that many

documents are relevant to a query although they do not contain any of the terms in the query. Consider a request for documents about “cheap furniture”; a document with nothing but “inexpensive table” may well be deemed relevant to the query. In Figure 1, we illustrate the real problem – the language (e.g.; common vocabulary) used by the document author is often quite different from the language used by the person entering the query. Often the person making the request must be prompted for more details or for some knowledge that will aid in the understanding of what is actually requested. A key problem in the realm of information retrieval is therefore to find some way to make that leap from the language of the person entering the query to the language of the document author in an automatic fashion.

Figure 1

Language of the User is different from the Language of the Author



Metrics

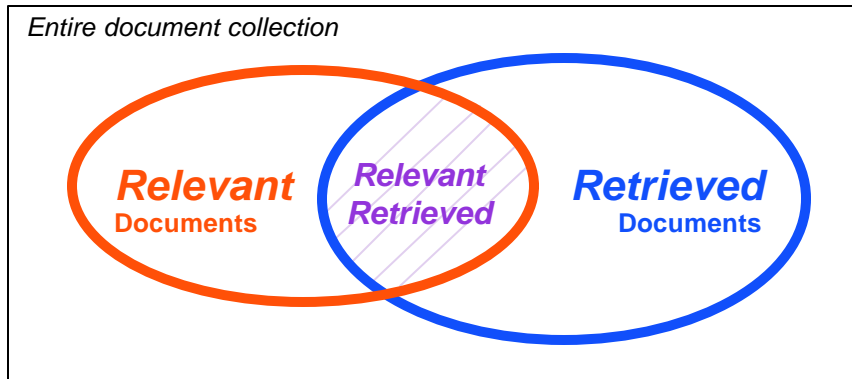
For measuring time, we have several measures, namely, *the time taken to retrieve the first, all, or some determined number of documents*. Clearly, we need another measure for identifying how well a retrieval algorithm is at finding documents.

Average Precision

Average precision is a measure widely used to evaluate information retrieval systems. Before people measured algorithms with something like average precision there was nothing but sheer speculation and self-proclamations. People might naturally think that “we can use a thesaurus to add synonyms to a query” and believe that it might “help” the accuracy of information retrieval systems. Until one can demonstrate scientifically that the average precision is improved then the notion that some technique either helps or hurts an information retrieval system must be simply referred to as merely a supposition or a hypothesis. We note that the average precision is not a perfect measure, but it does give us a means to test information retrieval algorithms and make some claims as to their effectiveness. In the information retrieval literature, efficiency is used to refer to how fast a system retrieves documents and effectiveness is used to refer to how “relevant” the documents are in relation to the query.

Average precision is computed by measuring both *recall* and *precision* at various points of recall. *Precision* is the ratio of relevant retrieved documents to retrieved, while *recall* is the ratio of relevant retrieved to relevant. Figure 2 illustrates these sets.

Figure 2
Measures of Accuracy



$$\text{Recall} = \text{Relevant-Retrieved} / \text{Relevant}$$
$$\text{Precision} = \text{Relevant-Retrieved} / \text{Retrieved}$$

Computing Average Precision

To compute average precision we need an answer set. Assume we retrieve the top ten documents in response to a query (see Figure 3). Now further assume that we show them to some judge and that judge deems that five of them are relevant. We might now be inclined to say that our system is 50% accurate. Well, this might work except for one little problem. The little problem is that there might only have been five documents relevant to this query *in the entire collection*. Finding all five is a wonderful achievement and should not be branded a mere fifty percent success. Hence, we might define some other measure as the ratio of relevant documents to the total number of relevant documents in the entire collection. For this measure, our example would be five out of five and deserve a 100% score. This measure is commonly referred to as *recall* while our first measure, the ratio of relevant retrieved to the total retrieved is called *precision*.

Figure 3: Sample Result Set

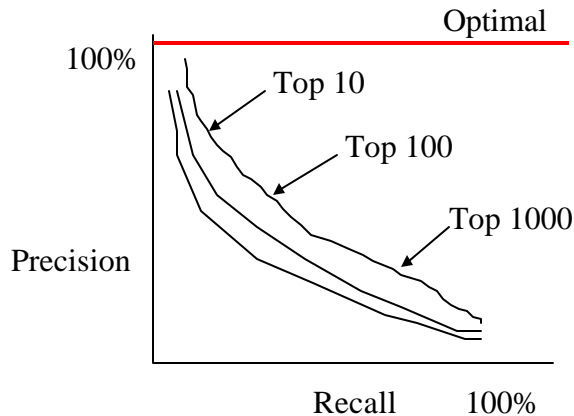
Only documents in red are relevant. Consider this answer set:

D1
D2
D3
D4
D5
D6
D7
D8
D9
D10

Since we have retrieved ten documents and gotten five of them right, precision is 50 percent. Recall is the ratio of five to the total relevant in entire collection.

A typical recall-precision graph is shown in Figure 4 with recall on the x-axis and precision on the y-axis. A recall of 0.10 indicates that 10 percent of all relevant documents to the query were found. Typically, we must retrieve more and more documents to find all of the relevant documents. As we do this, precision falls as our efforts to find that last relevant document usually fail, and we find a large number of non-relevant documents. Hence, the recall-precision graph usually resembles that curve as shown in Figure 4. We note that an optimal system would follow the straight line at the top of this figure. For this line to happen, we must *only retrieve* relevant documents. Wouldn't it be nice if while logged onto a web search engine, all queries entered returned only relevant documents and nothing else? That is the ultimate goal of the field of information retrieval. It should be clear that we are quite far from achieving this goal.

Figure 4:
Recall-Precision Curves



Average precision is computed by averaging the precision levels at various points of recall. 3-pt average precision chooses 0.33, 0.66, and 1.0. 11-pt average chooses 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0. Clearly, we cannot compute precision at 0 documents, so this point is often interpolated from the others. The 11-pt average gives us a single number that can be used to characterize an information retrieval system.

It is not easy to compute this measure. To do this accurately, the entire document collection must be manually examined to determine which documents are relevant to a query. Some test collections provided by the National Institute of Standards and Technology and used at the Text Retrieval Conference (TREC), rely on human judgment to assess documents retrieved from a variety of systems for the same set of queries. The assumption is that when all of the results are combined from all the participants at TREC, this union of results contains the set of all documents relevant to the query. This

approximation of the total set of relevant documents is called *pooling*. Clearly, this is not always the case. It is possible for some documents to be relevant but not retrieved by any of the participating information retrieval systems. However, this approach is a reasonable approximation of the set of all relevant documents and is often used to assess the effectiveness of information retrieval systems.

Another problem with using a standardized test collections that rely on human judgment for document relevance is that beauty is, actually, in the eyes of the beholder. Thus, what may appear relevant to one judge might not appear relevant to another. Some studies have shown a typical correlation between two users viewing of documents as matching only 80% of the time [xxx]. Although human relevant judges and the problems inherent in pooling do result in some level of fuzziness in the use of average precision – it is the most commonly used effectiveness measure and has been so for most of the last ten years.

With SimpleIR, we focus both on effectiveness and efficiency techniques. Our basic goal is to take at most a few seconds to respond to a query over several gigabytes of documents and retrieve a set of fairly relevant documents.

Information Retrieval Resources

The field of information retrieval is constantly changing as researchers around the world are always developing new algorithms. These algorithms are published in journals such as the *Journal of the American Society for Information Science and Technology*, *Information Retrieval*, and *ACM Transactions on Office Information Systems*. Annual conferences that focus on information retrieval are the ACM SIGIR (Special Interest Group on Information Retrieval) and ACM CIKM (Conference on Information and

Knowledge Management). The Text Retrieval Conference sponsored by the National Institute for Standards and Technology (NIST) contains results from users running on the standardized document collection described in this chapter. Its proceedings are published on the web at nist.trec.gov.

Book Organization

The remainder of the book is organized as follows: First, in Chapter 2, we describe the high level architecture of a search engine. We then begin with Major Section I where we describe a simplified information retrieval system in Chapters 3, 4, and 5. Chapter 3 is dedicated to a simplified parser, Chapter 4 to small-scale inverted indexing, and Chapter 5 to query processing. In Major Section 2, we continue with a description of a more sophisticated search engine. In Chapter 6, we describe advanced parsing techniques (e.g.; asynchronous parsing, part-of-speech tagging, web crawling, phrase processing, and information extraction), and in Chapter 7, we present advanced inverted indexing (e.g.; inverted index compression, algorithmic tradeoffs when choosing an inverted index construction algorithm, etc.). Chapter 8 consists of advanced query processing techniques such as query expansion with relevance feedback, thesauri, and semantic networks. Additionally, techniques to improve query efficiency such as the use of posting list and query term thresholds are described. In Chapter 10, we describe future topics in information retrieval such as integration of structured data and text, multi-lingual processing, and the processing of voice-transposed data. We conclude the book in Chapter 11.

Exercises:

1. Run SimpleIR for the sample documents called sample.txt provided in the CD found in the back of the book. Create an index and then try ten different queries.
2. Do you detect a pattern in what is driving performance of the system? You should see that some terms in a user query cause the query to run slower than others. What is the cause of this?
3. Modify SimpleIR so it accepts queries from the command line in the form SimpleIR -I <input file> -O <output file> where -I indicates an input file consisting of several queries and -O indicates an output file which consists of document identifiers and the relevance score assigned to each document. This will get you ready for testing a variety of similarity measures and will make it so you can avoid dealing with the user interface for the rest of this book. This book is not about how to build a good user interface for an information retrieval system – truthfully that is a topic for a whole other book on human-user interface.
4. Consider a query that contains five terms and retrieves 20 documents. Assume the query has 10 relevant documents and 6 of these are found in the top twenty retrieved. Document number 1 is very relevant, document 2 is kind of relevant, and the other four documents are somewhat relevant. However, the relevance judge has deemed all six to be relevant.
 - a. Compute the precision for this query
 - b. Compute the recall for this query
 - c. Compute the average precision for (.2, .4, and .6) points of recall.
5. Think of a query and a document that matches the query. Now think of using a thesaurus to improve the query. Give an example where using a thesaurus would dramatically improve effectiveness. Now give an example where a thesaurus would dramatically degrade effectiveness.

