Modern Information Retrieval

the concepts and technology behind search

Second edition

Ricardo Baeza-Yates
Berthier Ribeiro-Neto

Modern Information Retrieval

The Concepts and Technology behind Search

Ricardo Baeza-Yates
Berthier Ribeiro-Neto

Second edition

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

Chapter 15

Enterprise Search

by David Hawking

15.1 Introduction

Like libraries, corporations, government agencies, and not-for-profit organizations have to deal with documents in many different media and formats, but much of that information is unique and proprietary to the organization. Some of an organization's information assets may be held in relational databases or specialized applications, but much is unstructured text of the type information retrieval systems have been designed to deal with.

The application of information retrieval technology to information finding within organizations has become known as *enterprise search*. Enterprise search may be interpreted [719] as search of digital textual materials owned by an organization, including search of their external Web site, company intranet, and any other electronic text that they hold such as email, database records and shared documents.

It is quite common for users of enterprise search tools to compare their experiences unfavourably with those on the Web. "I can easily find my great grandfather's birth certificate among the fifty billion pages on the Web. How come I can't find last year's financial reports within my own small company?" Sometimes the criticisms are quite strident and are made even though the technology in use may be derived from a Web search engine and sold and supported by the same company. One of the aims of this chapter is to outline how the enterprise search problem differs from Web search.

This chapter describes the architecture of enterprise search systems and the function of the different components of such systems. It also discusses attempts to study real enterprise search activity in order to characterise it for the purpose of scientific study. Other topics include enterprise search evaluation methodologies, studies of enterprise search within TREC, tuning of enterprise search systems, the interaction of publishing and search, and the performance levels it is reasonable to expect of an enterprise search deployment. Two topics which are of particular importance to

enterprise search, though by no means restricted to it, are also covered: federated search and search contextualization/personalization.

15.1.1 Characteristics and Applications of Enterprise Search

Many characteristics of enterprise search represent a significant challenge for IR system designers [274, 1162]. Information in the enterprise may be structured or unstructured. Documents are produced by a variety of sources, perhaps in many different languages, and generally without formatting standards. Metadata may be created according to a number of different schemes, or may not be added at all. Not all users have the same access rights to all information, and some information, such as employee records, are highly confidential. The need to federate across different repositories of information means that a single ranked list must be created for data from a variety of sources and formats. Different contexts may require different ranking methods. That is, search tools for the enterprise must perform many functions in addition to simple indexing and query processing. A basic search tool will not do.

There is an expectation, based on experience with the Web, that finding corporate information should be fast and efficient, and that it should be done through a single interface. However within organizations these expectations have typically not been met, and there is evidence that employees spend a significant amount of their time searching for, and often failing to find, information needed to perform their work. For instance,

- According to IDC (International Data Corporation), a company with 1,000 information workers can expect more than $5M in annual wasted salary costs because of poor search. They report that people spend 9 – 10 hours per week searching for information and are not successful from one-third to half of the time [805].
- According to Butler Group, as much as 10% of a company's salary costs are wasted through ineffective search [526].
- A 2007 Accenture survey of 1,000 middle managers found they spend as long as two hours a day searching for information and that more than half the information they find during searching is useless [713].

Another area of high financial impact for enterprise search is the area of e-discovery. Search tools capable of auditably searching all the sources of information within an organization are in increasing demand for supporting discovery actions associated with high-value lawsuits [1378], even if these searches are conducted by external professionals. Further, effective search on an organization's outward-facing Web sites can be critical to an organization's mission – be it disseminating information, supporting government campaigns, matching job applicants to job vacancies, or generating on-line sales. E-commerce sites are often driven by a search tool whose functions include supporting search for product information and reviews, location of the actual product purchase page, search-driven advertising and intelligent recommendations. Consequently, enterprise search software on external Web sites is a highly valued source of information about the interests of customers and stakeholders. Further, query data they generate can give information about trends and sudden

spikes in customer/community interest as well as identifying unmet demand. Indeed, leading enterprise search tools provide extensive reporting capabilities.

Enterprise search tools perform other functions too. Navigation links, RSS feeds, faceted browsing and taxonomy displays on organizational Web sites are often powered by search engines. Search tools are increasingly used within organizations to locate expertise when putting together project teams. Further, automatically generated internal reports may include summaries derived from search results. Finally, the benefits of Web publishing and search are taken advantage of on almost all corporate intranets.

Enterprise search almost inevitably includes a small-scale Web search dimension. Upstill *et al.* [1619] showed that anchor text and PageRank variants were very effective in small-Web contexts, although simple in-link counts achieved most of the benefit of PageRank. Hawking *et al.* [720] investigated whether links from outside an organization could be used to improve the quality of Web site search for that organization. They found that most external links to the organizations under study tended to reference the site entry page and only a small number of additional targets. Thus, their value in answering specific within-site questions was negligible. Hawking and Zobel [724] studied the value of topic metadata in answering queries submitted to the Web site of an organization with a commitment to metadata mark-up. They found that topic metadata was of extremely low value in answering the queries, due to both inherent limitations and to poor implementation (despite resources committed). While there has been some technology transfer from Web search to enterprise search, the two applications differ in several significant ways, which are outlined in this chapter. One of the major differences is that, within an organization, there is no financial reward for spamming!

15.1.2 Enterprise Search Software

Enterprise search software has been available for some time, and some of the earlier systems were spin-offs from academic research in information retrieval. Companies such as FAST Search & Transfer[1] (acquired by Microsoft in 2008) and Autonomy[2] (which acquired Verity in 2005 and the Interwoven content management system (CMS) technology in 2009) are well-known for their enterprise search. Major search and software companies like IBM, Oracle, and Google have also developed products intended for this market. Google's Search Appliance is popular due to its ease of use and the familiarity of the Google name [70].

Smaller companies offering enterprise search products may create a niche through special features. Vivisimo,[3] developers of a search engine which provides clustered output, also have an enterprise search product for the corporate market. Endeca[4] offers a product with "guided navigation" in which possible search filters are offered as part of the results screen. Funnelback[5] specialises in "software as a service" (SaaS) for enterprise, Web site and portal search.

[1] http://www.fastsearch.com/
[2] http://www.autonomy.com/
[3] http://vivisimo.com/
[4] http://endeca.com/
[5] http://funnelback.com/

15.1.3 Workplace Search

It is useful to make a distinction between enterprise search and other types of search conducted by employees while they are at work. Most employees have access to a "desktop search" facility either built into their PC's operating system or supplied by *e.g.* Copernic[6] or Google.[7] Dumais *et al.* [518] report an extension of this type to include search of all the documents (*e.g.* downloaded Web pages, received email messages, etc.) previously viewed by the user.

In general, we can include all the searches conducted by employees under the label "workplace search". This label covers not only search of enterprise information, information held on the desktop, and information previously viewed, but also search of information sources held external to the organization, such as the Web, patent databases, legal resources, subscription information services, etc.

Since the set of sources to be searched varies from one employee to another, and since it is not feasible for an organization to build a combined index of all the relevant information, the only feasible single-search-box approach to workplace search is "personal metasearch" [1580, 1583]. A pilot survey reported in [1580] illustrates the diversity of sources accessed by different employees.

15.2 Enterprise Search Tasks

Organizations vary considerably in the degree to which unstructured information, and the need to conduct impromptu searches of it, is important to their business. At the low end, a concrete fabrication business or a hairdressing salon may have very little need for enterprise search. At the other end of the spectrum, search of internal and external information, both unstructured and semi-structured may be critical to the productivity and competitiveness for a firm of policy consultants. Obviously, it is crucial to the operation of national intelligence agencies. In a technical support centre, search effectiveness (of documentation and customer histories), can determine productivity and profitability.

15.2.1 Examples of Search-Supported Tasks

Many tasks carried out by employees are either made possible or made more efficient by search tools. Sometimes the search is supported by an enterprise-wide search facility but in other cases the search is embedded in a specific application. We now present some examples of tasks which may be supported either by application-specific or by integrated information retrieval tools. The list is far from complete but gives an idea of the range of applications which may be encountered.

Approving an Employee Travel Request

In order to decide whether to approve a travel request, a manager requires a variety of information: At what level of seniority is the employee? How beneficial is the event likely to be to the employee and the company? How much has the employee

[6] http://www.copernic.com/en/products/desktop-search/
[7] http://desktop.google.com/

spent on travel in the past year? What is company policy on this type of travel? Is the employee performing well? Would their proposed absence of work cause a loss of production or the failure to meet a deadline? A newly appointed manager in this circumstance would need to search a variety of information sources such as email, the HR database, and the policy section of the intranet, in order to make the right decision.

Responding to Calls in a Call Centre

Many call centres rely on efficient search tools operating over carefully prepared documentation to minimise operating costs. If the search tool always finds the right answer page, then a less skilled and less trained workforce can do the job for lower wages. If the search tool reduces time wasted looking for answers, then support calls are shorter and more calls can be handled with the same number of operators.

Responding in the Course of a Dispute

When projects fail or mistakes are made an organization may need to follow the trail of communication leading up to the adverse event, in order to counsel or discipline an employee, or to decide what position to take in negotiating with external parties. Effective search for critical emails and project documents may be critical to making the right response.

Writing a Proposal

For a private company, responding to a large "Request for Proposal" or "Request for Tender" opportunity can be a time-consuming and costly business. Many such RFPs require responses to hundreds of questions, and result in a proposal document exceeding a hundred pages. The cost of responding can be significantly reduced if a search tool can quickly and accurately locate the best response paragraphs and images from previous RFPs and also locate other useful and current company documentation.

Obtaining and Defending Patents

Industrial companies like Dupont, BASF, and Pfizer are fundamentally dependent on their patent portfolios. Before investing billions of dollars in a factory to manufacture a new chemical, product or drug, they must ensure that their intellectual property (IP) foundation is secure. Industrial companies typically subscribe to commercial patent database and literature services and use specialist patent search tools. Patent search poses many challenges, including: obscure language of patent attorneys; the need to search patents in all languages; the need to search for diagrams and chemical structures as well as text; the need to recognize variants of chemical and biological names; and the need to impose relational constraints over factors such as reaction temperatures. Information about searches and the areas in which they are being conducted is of course highly confidential information as it would be valuable to investors and competitors. IP searching takes several forms:

- Patent landscaping: Identifying patent gaps in a particular field in order to target the company's research into fruitful areas.

- Freedom to operate: Does a technology created by the company violate any patents held by others?
- Novelty search: Is a new discovery potentially patentable?
- Patent invalidity search: Can we discover prior art in a field which would enable us to strike down a patent held by a competitor which is impeding our business?

Selling to an Existing Customer

The probability of making a successful pitch to a customer can be substantially increased if:

- the pitch is targeted at solving problems the customer actually has,
- the vendor can present themselves as competent, professional and attentive to the customer's needs, and
- the vendor can identify who are the most useful contacts within the customer organization and what roles they play.

Successful customer relationship management (CRM) relies on the ability to effectively search, analyse and present all the data relating to that customer, including contracts, invoices, sales enquiries, email, and support requests. Selling to prospective customers can also benefit from effective search, but in this case the information to be searched lies outside the enterprise.

Expertise Finding

Expertise finding is a particular problem in large organizations. The need may arise during *ad hoc* problem solving or when attempting to put together a project team. In some cases, a dedicated software application maintains a register of expertise which can be updated and queried in normal database fashion. In other cases (see the CSIRO example below) information created and published for other purposes can be mined for expertise. In the latter type of system, identifying the set of candidate experts is a significant problem. It is easy enough to identify the email addresses in a set of Web pages which conform to the employee pattern, but how many of the extracted addresses correspond to people who have left the organization and how many of the rest, correspond to administrative or support staff rather than technical experts?

An early expertise finding prototype developed in CSIRO [440] intersected the crawl of Web pages with a current employee database. Passages of text including the name of a relevant employee or that employee's email address were extracted and added to a surrogate document named after the person. When an expertise query was processed against the experts collection, documents representing people were ranked and the contact details from the employee database entries for the top-ranked people were returned as the search results. Subsequent research within the TREC Enterprise Track has showcased improved methods including the language modelling approach of Balog *et al.* [131, 132]. In this context, Serdyukov *et al.* [1450] demonstrated benefit from accessing information from the external Web in identifying experts within an organization.

How results of an automated expertise finding system are presented may be critical to acceptance of that system, as a person's degree of expertise may not be reflected in the volume of subject-related text in documents available to the expertise finder. The name and contact details of a company's media liaison representative may appear on all the company's technical documents while a Nobel-prize-winning scientist may choose to have very little visible presence on the Web or in internal electronic documents.

Operating an E-commerce Site

Some businesses such as retailers, catering suppliers, travel agents and employment services rely on e-commerce Web sites for some or all of their revenue. A typical e-commerce site provides product search, coupled with query suggestion, faceted navigation and automatically generated cross-sell recommendations. Ranking algorithms for e-commerce sites must take into account a variety of non-traditional factors such as: stock levels, use-by-dates, and profit margins on different products; and whether items are "on-sale" or part of some promotional campaign. E-commerce sites are sometimes custom-built database applications, but they may also be built on enterprise search tools with the relevant capabilities. Endeca, Autonomy and FAST are well known in this space.

15.2.2 Search Types

Broder [268] identified three distinct types of Web search: navigational, transactional and informational (see section 7.2.1). Queries of all three types may be submitted to enterprise search engines, *e.g.*

navigational: 'library', 'HR', 'plastics division'.

transactional: 'buy parking permit', 'renew library card', 'claim expenses'.

informational: 'IP policy', 'customers in Spain', 'product xyz - error 57'

Many sub-types of these categories are to be found in the search-based tasks exemplified in the previous subsection.

15.2.3 Studying Enterprise Search

It is very difficult to study search behaviour within an organization of which you are not an employee. In general, organizations do not want their competitors to know what their employees might be searching for, or even what sets of documents they might be searching. For this reason, query logs are unlikely to be made available for publication, or even for perusal by external researchers. In any case it may be very difficult to infer task from knowledge of queries submitted.

For similar reasons, it is generally unlikely that experimenters will be permitted to follow employees around with a clipboard and record their search behaviours. Not only that, but the presence of an observer may change behaviour. Finally, when search is only a small part of an employee's activities, the amount of experimenter

time required to gather useful data will be too large to be affordable. Despite these problems, search behaviour within organizations has been studied by Hertzum and Pejtersen [756] (engineers), Hansen and Järvelin [697] (search within the Swedish Patent Office), and Freund and colleagues [590, 589] (software engineers).

For the TREC Enterprise track in 2007/8 [126], Science Communicators from Australia's government research organization CSIRO[8] provided information need statements and "relevance" judgements for two real tasks arising in communicating CSIRO's science to the public and to potential partners. The following are paraphrases of the tasks studied:

1. "I'm writing an overview Web page to CSIRO's research in an important area, *e.g.*, dry land salinity. Find me a set of important Web pages within CSIRO which would be good candidates for linking from the overview. For example, pages describing significant CSIRO projects in the area, reports, software tools, maps and data sheets which may be of use to partners or the public."

2. "For the same overview Web page, trawl through the CSIRO Web content and identify CSIRO's experts on that topic. You can make use of the fact that CSIRO email addresses have the form `firstname.lastname@csiro.au`."

15.3 Architecture of Enterprise Search Systems

Organizations vary by many orders of magnitude in the number of unstructured documents which they publish internally. Obviously, there are organizations with almost no shared electronic text, while the IBM intranet was reported in 2003 [541] to contain around 50 million documents.

As the scale and complexity of internally published material grows, the importance of efficient and appropriate index-building workflows increases. Figure 15.1 sketches the three broad phases involved in building a unified index of heterogeneous enterprise data: gathering, extracting, and indexing, as we now discuss.

15.3.1 Gathering

The gathering phase, which corresponds to crawling for Web search engines, as discussed in Chapter 12, may be arbitrarily complex. First, maintaining coverage and freshness of crawled internal Web data can be subject to many of the challenges faced on the external Web – redirections, publication of multiple copies of the same content at different URLs, difficulty of identifying content which has changed recently, near-duplicate detection and network bandwidth issues (*e.g.* between offices in different countries or cities), difficulty of link extraction from JavaScript and Flash. Note however that, within an enterprise, techniques for reducing the cost and duration of crawling by allowing servers to supply lists of changed content (or even partial indexes) may be deployed without risk. In addition, apart from the issue of permissions, and the need to efficiently identify recently changed content, scanning of file systems is relatively straightforward. Further, careful crafting of SQL queries may allow the extraction of all and only the useful information needed in database gathering.

[8] http://csiro.au/

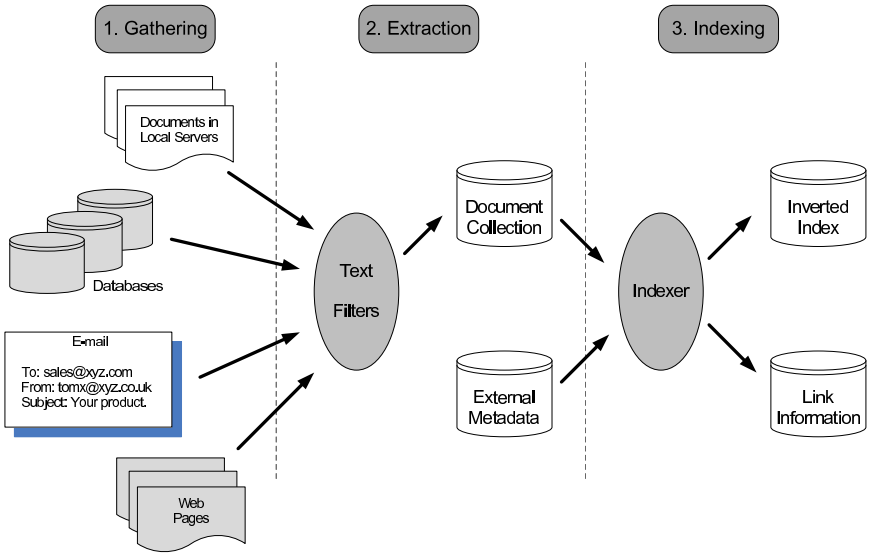

Figure 15.1: Gathering and indexing workflow.

Second, successful gathering from the plethora of enterprise software applications which may be deployed at an organization, depends upon the availability of appropriate APIs or adapter software. So-called "enterprise software", records management (RMS, EDRMS, ECM) systems, customer relationship management (CRM) systems, and content management (CMS) systems are some of the generic classes of such systems, and each class includes many competing proprietary systems. Particular issues arise with systems such as Lotus Notes, in which objects of various types (*e.g.* forms, views, documents, navigators, and agents) are accessible both through a native API and through standard Web publishing (Domino). On the one hand, a single "document" may be synthesised from multiple content fragments while, on the other, multiple views of the same basic content may be published at multiple URLs. To illustrate, all of the following URLs from an anonymized organization actually represent the same document.

- .../d/xyz\%40.nsf/mf/3240.1?OpenDocument
- .../D/xyz@.nsf/b06660592430724fca2568b5007b8619/1c87d9876bc11ee8ca256fd5007722a8!OpenDocument
- .../D/xyz@.nsf/5087e58f30c6bb25ca2568b60010b303/1c87d9876bc11ee8ca256fd5007722a8!OpenDocument
- .../d/xyz@.nsf/w2.2.2/1c87d9876bc11ee8ca256fd5007722a8!OpenDocument
- .../d/xyz@.nsf/w2.2.1/1c87d9876bc11ee8ca256fd5007722a8!OpenDocument

Gathering content objects from a database may avoid indexing duplicate content, but in general it will make it difficult to generate URLs suitable for presentation in search results. Furthermore, annotations such as tags and anchor text, and user

behaviour data (which can contribute significantly to good document rankings) are likely to be associated with URLs rather than with content fragments.

Third, in many applications it is necessary to gather access control lists (ACLs) and external metadata (*i.e.* information about a document which is recorded in a separate database or register) applying to gathered documents. Fourth, whether email can be made searchable to other than the senders and recipients of each message requires either: a corporate decision that email sent to an organizational address is not private to an employee, or the implementation of a system for segregating organizational and private email. In some organizations, every incoming message is archived and potentially searchable; in others email search would be restricted to databases maintained by post-office software such as Exchange; in yet others search is only possible over personal mailboxes, leading to hand-over problems when staff leave.

Fifth, some organizations have enthusiastically embraced and adapted so-called "Web 2.0" approaches for their employees and even their customers, such as folksonomy tagging, FaceBook-style[9] social networking, blogging, instant-messaging, and "twittering".[10]" According to a pharmaceutical company which has enthusiastically adopted the latter two technologies and integrated them with SMS messaging on mobile devices, searching "information flows" is a much more critical need than searching repositories.

Sixth, the gathering process may take a very long time and generate significant additional telecommunications charges for an organization. For example:

> Scenario 1: An Australian government agency has offices in all nine capital cities. Staff in each office create documents on a locally shared file system. The organization wishes to provide unified search over all nine file systems, but the bandwidth of the links between offices is similar in speed to dial-up modems. To increase the bandwidth under their outsourcing contract would incur significant additional costs. Query submission rates are low.

In circumstances such as those in Scenario 1 (abstracted from a real case), the cost of frequent gathering may not be considered to be justified by the volume of queries, and a federated (metasearch) approach might be preferred.

In all modes of gathering, very large efficiency gains can usually be achieved by taking an *incremental* approach. In a large intranet or database, it is unlikely that even 1% of content will change in the course of a day. It would probably not be feasible or cost-effective to gather full content every day for a 50 million page intranet, but quite feasible to gather the changed content if that can be identified without a full scan. Note that gains from incremental gathering may flow through to filtering and indexing.

15.3.2 Extracting

Extracting (or filtering) text and metadata from binary documents such as PDFs and Office documents (see Chapter 6), seems as though it should be a simple engineering exercise. In practice, filtering issues may be a major cause of user dissatisfaction with

[9] http://www.facebook.com/
[10] http://twitter.com/

enterprise search results. This is because failures in filtering may result in meaningless titles, poor quality "cached copies", and garbled summaries. Further, critical documents on a topic may not even be recognised as matching the query.

Why is filtering harder than it seems? The first reason concerns the use of proprietary document formats, such as those discussed in section 6.5. Some such formats are unpublished and obscure, thus making reverse-engineering difficult. Sometimes details change with every new release of the document creation software, thus increasing the number of different formats which must be supported by third-party filter developers. The adoption of compressed XML formats by OpenOffice and by recent versions of Microsoft Office is potentially a major step forward for ease and accuracy of filtering.

A second factor concerns the loss of text semantics when encoding a document in a presentation-oriented format like PostScript or PDF (portable document format). At creation time, most documents are represented in reading order with special sections such as titles, headings and tables explicitly marked. When converted to PostScript, the basic operations are graphically oriented: *e.g.* "draw a gray line of weight 0.1 from point (x_1, y_1) to point (x_2, y_2) in the current coordinate system", "print the n-th character from the current font table at position (x, y)". The reverse transformation, from graphic to text space, is in general hard, time-consuming and error-prone.

A third factor concerns the representation of metadata. Many well-known formats such as MSWord, PDF, OpenDocument, and JPEG are capable of storing metadata such as title, author, subject, and date. In practice, however these metadata is usually missing. If present, it is frequently of low value. Perhaps if internal document authors were able to see improved searchability arising from good title, author and date metadata, the situation would improve.

In any case, many commonly used document formats are limited in the types of metadata they can record. This leads to reliance on external metadata repositories for recording details about the document, or alternatively, to the absence of certain metadata which might be useful in retrieval.

> Scenario 2: An organization creates its reports for external Web publication using Microsoft Word. So that they can be read and printed with consistent pagination by all their stakeholders, they convert them to PDF. Unfortunately, employees seldom think to record titles in the properties of the Word documents. Word uses the file name by default. When converted to PDF, the phrase "Microsoft Word" is added to the file name and stored in the PDF metadata section, to be picked up by the search engine's PDF filter. When a customer searches the reports section of the site, all of the results have titles in the same pattern: MicrosoftWord-fileXX.DOC. Many searchers perceive that the results are probably near-duplicates and that they are in Microsoft Word format, although they are actually PDFs.

Fourth, textual information may be present in a document only in scanned form. Many offices are now equipped with photocopier/fax/scanner devices which scan printed documents and send resulting PDFs to specified email addresses. Text extraction from such PDFs requires the use of OCR software, increasing both elapsed times and error rates. A fifth factor concerns the accessibility of content within a document, which may be compressed in a variety of schemes and may be encrypted.

PDF documents may be flagged internally to prohibit the extraction of text. Finally, important structure within a document may be represented in many types of document by typesetting conventions. For example,

> Scenario 3: An organization requires that employee profiles must be typeset as follows: "Start with surname first in 12-point Times-bold, followed by forenames in 12-point Times-Roman. Next line contains employment classification, then years of service ..."

The performance of a retrieval system working with data created along the lines of Scenario 3 can be substantially improved by using scraping techniques to recreate the logical fields, prior to indexing. Unlike the Web, within a single organization it is reasonable to expect only a small number of different conventions of this type.

Filtering of large collections of binary format documents may be very time consuming. The time taken can usually be dramatically reduced by incremental filtering, where the only documents to be filtered are those which have changed since the last update.

15.3.3 Indexing

There is no particular reason for index formats used by enterprise search tools to differ from those used on the Web or elsewhere. Inverted indexes (see [1798]) are commonly used, with additional structures for representing textual annotations (see section 15.3.4) and static scores. In enterprises, however, there is a particular need to index fielded data (distinguishing types of metadata from each other and from document content). Indexing systems vary in how well they support the different types of data to be indexed, the rate at which they can index data, their ability to efficiently support phrase and proximity operations and the compactness of the indexes they produce. Appendix A presents a comparison of open source indexing software on these dimensions.

A challenge in designing an indexer lies in how best to deal with incrementally updated content. An incremental indexer deals with updates as follows: New documents are dealt with by appending a new entry to the document table and new entries at the end of the postings lists corresponding to each of the terms it contains. This potentially requires a lot of random-access I/O and may be in conflict with the document ordering necessary to support efficient document-at-a-time (DAAT [273, 1046]), query processing. There may be no room for a new entry at the end of a postings list, with the consequent need to free the space for the current list and create a new one at the end of the inverted index.

Deletion of documents from an incremental index poses even more problems, particularly when the postings lists are compressed. A typical solution is to leave the postings where they are but to mark the document as deleted. Updating of a document can be treated as deletion followed by insertion. Over time, an incremental index can grow significantly in size while access speeds decline due to fragmentation and loss of locality.

An alternative to using an incremental index is to maintain a combination of baseline and update indexes and search them in parallel, as illustrated in the following abstracted scenario:

Scenario 4: A media organization maintains a Web site of several million documents. It indexes the entire site every weekend to create a baseline index. Each night, an update index of all the new content created since the baseline is made. Documents in the baseline which have been updated are marked as "killed". Every 20 minutes, the content of the News part of the site is indexed. By default, searches on the site query a meta-index with three components: baseline, update and news.

Many variations on these two methods for updating indexes are possible. For example, a superstructure can be created so that the combined baseline / update indexes appear as a single index. Alternatively, a tool may be provided to merge baseline and update indexes into a single index.

15.3.4 Indexing Textual Annotations

As on the Web, enterprise documents (including non-HTML documents) may be annotated using various mechanisms: anchortext from links, officially applied metadata, click-associated queries [721, 1734] and folksonomy tags [1131]. Figure 15.2 illustrates how such annotations can be indexed for use in query processing. They can be used to provide both query-dependent and query-independent scoring components. Aggregated annotations may be scored separately and combined with text scores [1225] or they may be treated as fields of the original document [1370].

Naturally, organizations vary considerably in what types and volumes of annotation data are present. Few organizations support folksonomies but some are experimenting with them. The PowerHouse Museum[11] in Sydney, Australia allows visitors

[11] http://www.powerhousemuseum.com/

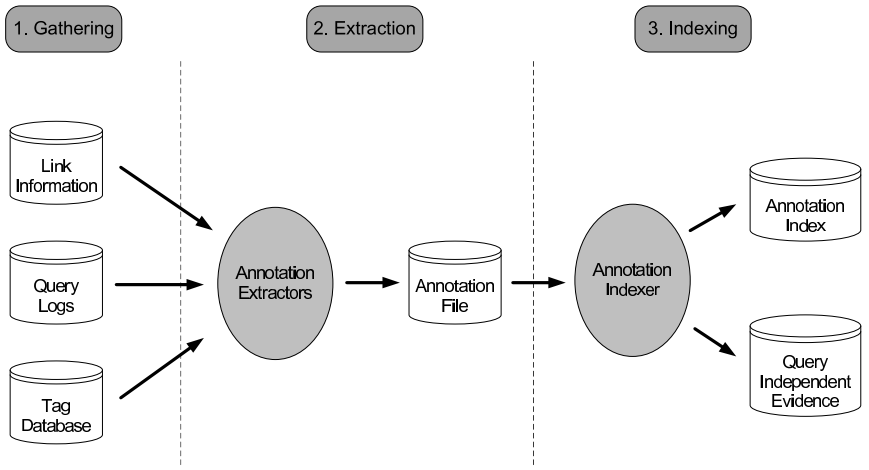

Figure 15.2: Processing of annotations to support effective enterprise search.

to their external Web site to apply tags to the Web pages representing their exhibits. Experiments with folksonomy tags have been conducted in IBM but, as reported in [502], volumes of data were small.

15.3.5 Query Processing

As noted in section 15.5, queries often fail in enterprise search because of a difference between the language of the query and the language of the documents. Enterprise search systems may provide tools such as thesaurus expansion, query suggestion, stemming, and relevance feedback to help bridge this gap.

As in Web search, more effective ranking within an enterprise may be achieved by combining text-derived scores with a static score. The static score formulation may need to be tuned to the characteristics of the particular publication environment. For example, in some organizations, link counts and/or URL lengths may not be correlated with probability of being a useful answer. Further, within an enterprise, there may be no inter-linking and the static score may take into account new factors. For example,

- Frequency of access to a resource.
- Recency of publication.
- Spam score of emails.
- Document type or genre. (Over all queries, some document types or genres may tend to be more useful than others.)
- Repository. (Perhaps bias in favour of results from the staff database over those from an email archive.)

In a library or e-commerce site, static scores may take into account the number of times an item has been borrowed or bought. In addition, benefit to the publisher (*e.g.* profit margin on an item, the perishability of goods or the availability of stock) may be considered in determining scores.

The relative value of query-independent contributors to the static score may vary enormously from organization to organization: Publication dates may be reliable in some organizations and not in others; The value of URL features is grossly diminished in an organization where URLs are generated in (usually arcane) fixed formats by a content management system; Normally the presence of thousands of links to a page would signal great importance or popularity, but not if they arise from a single navigation template, created by a single person.

Optimal search rankings must suppress the presence of near-duplicate results and encourage diversity, in order to overcome the likely presence of multiple drafts and versions of office documents, the presentation of sales materials in multiple forms, the presence of the same documents on file shares and in email attachments, and the universal problem of publishing the same material at multiple URLs. Note that it is generally not advisable to eliminate near-duplicates during gathering or indexing. Otherwise, a search which is scoped to a subset of the information assets may fail because the only copy of a wanted document has been eliminated because it is a near-duplicate of a document in another scope.

Heterogeneity is a significant cause of ranking difficulty when attempting to provide "single-search-box" access to all of an organization's information assets. Scoring the organization's Web documents may potentially use factors such as URL structure and length, link counts and anchortext that are not present in email, the staff database, the CRM system or the records management system (RMS). Is it possible to automatically generate a unified ranking in this environment which does a good job of ranking the Web assets and does not bias too strongly for or against them relative to other assets? Maybe a different type of presentation is preferred.

The optimum tuning of an enterprise-wide search system and the optimum way to present results may vary greatly across different individuals or different groups of employees. The reader is referred to section 15.6 for a discussion of this topic.

15.3.6 Presentation of Search Results

In many cases, the source or type of a search result may be used by a searcher as a signal of how useful that result is likely to be. Results in a unified ranking may include icons representing source or document type, or thumbnail images for products or staff profiles. Alternatively, the search result list may be segmented by source or type. For example results from the staff directory may be presented above results from the intranet, which in turn are presented above results from the external Web site. It is also very common within organizations to provide searches which are *scoped* to a clearly defined subset of documents within a comprehensive index. For example, the search box on a Human Resources intranet Web site may restrict search results to URLs from that site only. Similarly, other search interfaces may be provided to search only the staff directory, only policies and procedures, or only the staff bulletin.

In some cases, it makes sense to present results in an order other than descending probability of relevance to the query and/or to provide searchers with a choice of sorting options: E-mail search results may be sorted in date order, with most recent first; Publications may be sorted by file type first, then in alphabetic order of title. Finally, in an accommodation finding portal an element of randomness may be needed to avoid massive commercial advantage to one or two hotels in an area and corresponding disadvantage to the others. Similar considerations may come into play in an expertise finding application.

Enterprise search tools may provide a range of extra facilities to help employees or site visitors get maximum value out of a search they have conducted. Such facilities include: clustering, metadata facet counts (see the Location, Year, and Format facets in Figure 15.3, and see [738] for a comparison of clustering and faceting), multi-document summaries, spelling suggestions, and related queries. Enterprise search systems may also include tools for analysing a deep results set and extracting commonly occurring entities such as names of people, places and organizations, email addresses, phone numbers, and super-phrases of the input query. Search results may be shown in the context of a map (with the ability to search for documents tagged as being near the chosen results). As on the Web, enterprise search results for images and products should feature thumbnail images. Similarly, results for video segments should include key frame thumbnails. Further, users may register interest profiles to activate automatic alerts via RSS or email.

It often makes sense to integrate search into an enterprise application and to

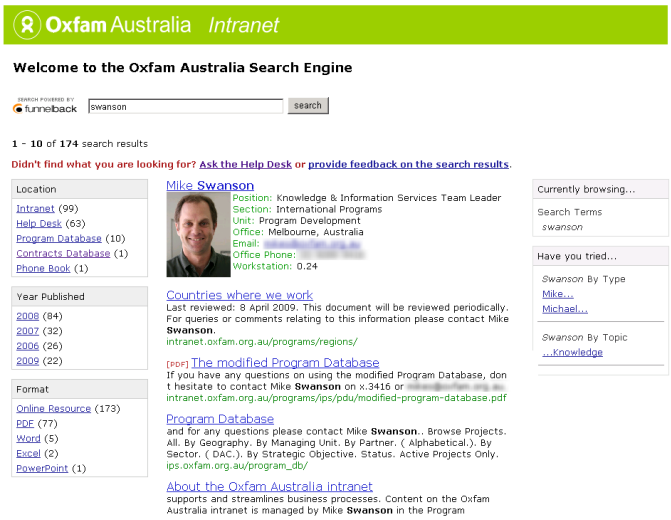

Figure 15.3: Screenshot illustrating enterprise search over a heterogeneous set of resources. Note the image in the result from the staff phone book, the PDF indicator, and the links on the left, allowing search to be narrowed to particular repositories, particular years, or particular formats. Note also the ability for users to provide feedback on the results obtained. Screenshot used with kind permission of Oxfam Australia.

present results in the context and format of that application. For example, email search results may be presented as a virtual email folder, supporting all the usual folder operations. As another example, a document processing application may be configured to continuously conduct searches automatically derived from text recently added to the document and to provide the results of those searches in a format which facilitates citing, quoting or incorporating in the document under construction.

Finally, enterprise search tools can play key roles in the customisable employee interfaces operated by some organizations along the lines of the *My Yahoo!* portal and the *Yahoo! Companion* toolbar described by Manber *et al.* [1076]. The idea that a portal page can be customised to the needs of an individual seems just as applicable within an organization as it is on the Web. A customised portal page can provide alerts, targeted summaries of the things that a particular employee needs to know about, action links, and personalized search facilities. Behind the scenes, a search tool can provide alerts, RSS feeds, and of course the tailored and scoped search (see section 15.6 for more detail on how search can be personalized). Manber *et al.* provide a number of lessons learned from *My Yahoo!* which have applicability within the enterprise. They highlight the importance of user interface design and insist that behaviour of the interface must be predictable by users. They warn that people generally don't understand the concept of customization and that most users do not take the effort to customise, concluding that a lot of effort should be put into optimising the plain vanilla version of an interface.

15.3.7 Security Models

In order to provide comprehensive search to privileged users, an enterprise search tool must be given omniscient access to all of their organization's information assets. The tool must therefore enforce (with 100% accuracy) the security of that information. This requirement necessitates a lot of careful engineering, due to the complexity of security models and the need to maintain accurate and efficient search. Rights to view documents or search results depend upon the user login (and on whether the searcher is logged in at all.) Threats which may arise from the use of an enterprise search tool include:

1. Unintended actions carried out when an internal crawler accesses an active server page or CGI script. For example, the system administrators at a university known to the author provided themselves with a `delete-page.cgi` facility with which they were able to remove another Web page through a simple browser request. They used it to remove a particular page which they later edited and reinstated. Unfortunately, when they accessed `delete-page.cgi`, the access was recorded in an online Web server access log which included links to all accessed pages. When a crawler next scanned the site it found the Web log page, and followed its links, leading to deletion of the newly edited version of the page!

2. Allowing someone to see content via the search engine to which direct access would be forbidden.

3. Preventing someone from accessing content via the search engine to which they would be granted direct access.

4. Providing means by which a malicious unprivileged user may deduce the existence of a sensitive document and possibly deduce something about its content. The most extreme example of this is when search results are shown but links to the actual documents are blocked. Potential sources of information about inaccessible content include hit counts, facet counts, multi-document summaries, clusters, and even response times.

5. Externally accessible enterprise Web site search is potentially vulnerable to cross-site scripting, JavaScript and other types of injection and to buffer overflow vulnerabilities.

Access to an organization's documents is generally controlled by mechanisms such as Access Control Lists (ACLs). These may specify which of a range of actions (*e.g.* reading, writing, indexing) are available to particular individuals or to groups. Folders or directories may also be subject to ACLs. The computation of whether a particular user may access a particular document can be quite complex, starting with the requirement that the user be authenticated, their group memberships ascertained, and their individual and group access rights checked against a chain of parent folder ACLs and the ACL of the document itself. Organizations may use a network authentication protocol such as Kerberos[12] and implement an enterprise single sign-on system to

[12] http://Web.mit.edu/Kerberos/

avoid the need for users to authenticate themselves to each of the different enterprise applications they use.

There are interesting differences between email messages and other documents from the point of view of security. The author (sender) of an email message specifies a list of recipients, but in most systems cannot specify an access control list for the message. Instead, copies of the message may be stored either in folders in the recipients' own file systems or in a central mail database. Access control to the message is thus determined by the recipients or by administrators of recipients' mail databases. Copies of documents attached to email messages may thus end up with very different access rights to those of the original. Access control may be possible only at the folder level, not on a per message basis.

Collection-level Security

Ideally (from the point of view of simplicity and efficiency) an organization's information assets can be simply divided into collections with uniform access rights. For example: a general-access collection, a finance collection, a senior-management collection, and an HR collection. Searches by employees are processed over the subset of collections appropriate to their role and per-document tests are not required when presenting results. Unfortunately, in most cases, the applicable security model is much more complex than this.

Document-level Security

Where a collection-level security model is not appropriate or just not adopted, access controls must be applied at the level of individual documents. Behind the scenes, a query submitted by a user results in an internal ranking over all documents, which must then be filtered, result by result, to exclude all and only the documents not accessible to the user, avoiding threats 2 and 3. To address threat 4, all other search results should be completely suppressed. A search tool operating in a secure environment should not present result snippets (however brief) representing documents which the current user is unable to view, nor should excluded documents be represented in any counts or result set analysis presented to the user.

Different organizations impose different requirements on the security restrictions which should be applied when searching. In the most extreme case, organizations may insist that current access controls should be applied on a per-document basis at the instant the search is conducted – late binding. If an employee's role changes at 5.00pm on a particular day then at 5.01pm on that day, they should be able to see all and only the documents appropriate to their new role.

Bailey *et al.* [127] illustrate the adverse effects on search response time of this extreme model, caused by the significant time taken to check access rights to each document in a large result set, particularly when the original documents are held on a non-local network domain. Less delay is caused if the search engine is permitted to cache access control data for the documents it indexes. In this early binding model, a person changing role may continue to have access to documents appropriate to their former role for a period after the role change, and may have to wait to gain search access to documents they are newly entitled to view.

15.3.8 Federation/Metasearch

When building a unified search across all the information sources (repositories) within an organization, it is sometimes not feasible for the enterprise search engine to gather, extract and index all the information from all of the sources. For example, the gathering processing from a particular source may be too slow, the network traffic too expensive or slow, or the size of the data too large. Such challenges may be faced in Web search, too [99]. Alternatively, the data may be locked into a proprietary application which provides no export facility (yes, really!). If the problematic source provides its own search facility, then it may still be possible to provide a unified search by taking an approach known in various contexts as "search federation", "metasearch", or "distributed information retrieval". More details on federated search are given in section 10.7 in the context of distributed Web retrieval and in section 11.10.3 in the context of Web metasearch.

If an employee wants their unified search interface to include their own personal (desktop) information and sources external to the organization, then metasearch is the only feasible solution [1580]. In metasearch, a query is received by a broker and forwarded to the search interfaces of the sources being federated. The broker then combines the separate result sets into a single set to be returned to the user. Pioneering work in this area was carried out by Gravano *et al.* [667], Voorhees *et al.* [1653] and Callan *et al.* [323]. Difficulties arise because the ranks and scores returned by the different search interfaces may be quite incompatible. The top ranked document from one source may be a much weaker match than the 50th rank from another which is more oriented to the topic. In the simplest case this arises because IDF values for some terms (see Chapter 3) may be subject to large variation across sources. In more complex cases, score variations may be due to different static weightings or annotation scores.

Search federation may pose particular problems for maintaining document level security. User credentials must be forwarded by the broker to the individual search services, which must be relied on to apply them correctly. A reliable single-sign-on mechanism across all the sources to be integrated seems almost indispensable.

Five Sub-problems of Metasearch

In general there are five problems to be addressed in a metasearch application:

1. At service definition time, *identifying* and choosing the sources to be federated. This may be a straightforward manual listing of well-known sources, or it may involve a scan with automated identification of search interfaces [424].

2. At service definition time, and as often as necessary during the operation of the service, *characterising* the sources – How many documents do they index? What is the language model of the documents indexed? How effective is the ranking algorithm employed?

3. At query time, *selecting* the subset of available sources to be included in the search. There is some evidence that optimal selection can outperform a search over all of the sources, but this is seldom or never achieved in practice. However,

selection may reduce costs incurred through network traffic, database subscription charges or per-query costs which may apply to particular external sources. In general, selection relies on source models built during the characterization phase.

4. *Translating* the query into the query language accepted by each of the federated sources.

5. At query time, *merging* the result sets returned from the search facilities at each of the sources.

A great deal of research has been conducted into problems 2, 3, and 5. Problem 1 has been relatively little studied, as the sources to be federated are usually a given. Problem 4 has received scant attention because it is usually either trivial or intractable – when search engines operate with different semantic models or support different sets of operators, accurate translation is not possible. For example, it is not possible to faithfully render a Boolean query involving negation, conjunction and disjunction as a simple bag of words. Nor is it possible to approximate truncation operators (wildcards) or regular expressions for a system which does not support them.

Unfortunately, the majority of work in the distributed information retrieval area has been evaluated using sources simulated by partitioning TREC Ad Hoc data. These partitions show much less variation in types and sizes of documents and collections than would be expected across federated enterprise repositories, and lack categories of information, such as document types or human interaction data, which might be useful in enterprise federation.

Sources to be federated may cooperate with the broker in various ways. For example, they may supply accurate statistics about collection size and document frequencies. In general, however, the sources to be federated in an enterprise or personal metasearch application will not cooperate with the broker. In the uncooperative case, it is necessary to sample documents via the search interface in order to characterise the server. Callan *et al.* [321] propose what seems to be a fairly *ad hoc* sampling method but which has been shown to work reasonably well. Subsequent work by Bar-Yossef and Gurevich [135] took a more principled approach to avoiding sampling biases, based on double rejection sampling. Four sampling methods were evaluated by Thomas and Hawking [1582] across a set of diverse collections intended to represent those which might be federated in a personal metasearch tool. The random walk method of Bar-Yossef and Gurevich was found to work better than alternatives but occasioned a very high cost. Thomas and Hawking proposed a more efficient *multiple queries* sampler which attained similar representative accuracy.

The multiple queries sampler submits a number of high-recall queries sampled from a pool derived independently of the collection, and requests k results, where k is as high as practical given the search engine. It samples documents from the union of the result sets. Queries which produce no results (underflow) and those which produce more than k results (overflow) are rejected. Using the union of result sets tends to help reduce the problem of *query bias* (the tendency of long, term-rich documents to be returned in response to many queries). Rejection of overflow queries avoids *ranking bias* (the increased probability of sampling documents with high static scores) and choosing a high value of k reduces the likelihood of rejection.

Estimating the size of a collection via its search interface generally relies on methods developed for estimating fish or animal populations. Instead of trapping and releasing animals, documents are sampled and re-sampled using methods like those described above. In the simple capture-recapture method, the number of documents in common between two independent, unbiased samples can be used to estimate the population size. Shokouhi *et al.* [1473] describe new methods in this area.

The number of distinct published methods for source selection now exceeds forty! Some rely on information obtained from the sources using probes such as the STARTS protocol [665]. Others assume the availability of term frequency data, while yet others assume no cooperation. The very well known CORI method of Callan *et al.* [323] treats each collection as a document and the set of collections as the collection and uses a standard relevance calculation as the basis for selection. In the uncooperative case, its analogues of TF and IDF (see Chapter 3) must be estimated from samples. Selection methods may take into account estimates of the effectiveness of the retrieval systems operated at each source, as the value of selecting a server with good answers is negated if they are unlikely to be found in response to a query [437].

Discussion of this topic would not be complete without mention of the decision theoretic framework proposed by Fuhr [600], which uses costs for retrieving relevant and irrelevant documents, expected retrieval quality, expected number of relevant documents at each source and costs for document delivery and query processing to make optimal decisions about how many documents to request from each source (possibly none). As previously noted, there is a dearth of research on selection methods within the context of enterprise source federation. One feels that in this context, better methods may be found which attempt to match the types of documents (such as email messages, calendar entries, contact details, service histories, technical manuals, etc.) held in particular repositories with the task behind the query.

As with selection, so there are many methods for merging results. Lawrence and Giles [987] proposed an effective merging method in which all the documents from the primary result lists are downloaded and locally ranked for relevance. Downsides of this method on the Web are the network traffic generated for each query and the delay in finalizing the merged result list. In an enterprise, these issues may be manageable, but other difficulties may arise. For example, ranking heterogeneous document types may pose difficulties and, even more importantly, proprietary applications may not deliver full documents. Rasolofo *et al.* [1335] propose and evaluate strategies for merging results in the case of a current news metasearcher, where sources vary greatly in the type and quantity provided for each search result "snippet". They were able to obtain results from the snippets which approached the performance achieved by downloading and locally indexing the full documents.

Presentation of the results of a metasearch is important to get right in a heterogeneous environment. Different types of result, such as images, client contact details, and corporate policies may need to be presented differently. Furthermore, there may be value to the user in clearly signalling the source of each result. One way to avoid merging problems is to avoid merging altogether, *i.e.* to present result lists for different sources in separate columns, as made prominent some years ago by the A9[13] multi-source search engine. Another alternative is to present results in a list which is segmented by source. Although the segments of the result list illustrated in Figure

[13] a9.com

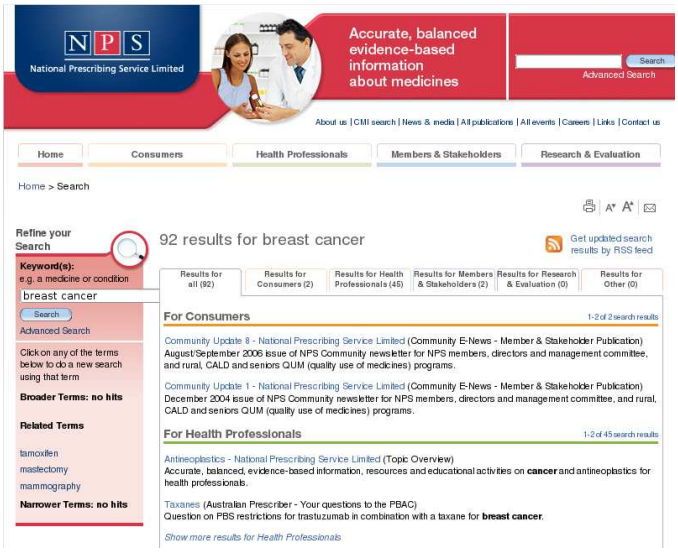

Figure 15.4: Segmented result list presentation, in this case designed to accommodate disparate groups of visitors to the site. A searcher can click on a link to expand results in a particular category. Screenshot from the National Prescribing Service.

15.4 don't actually correspond to separate sources, they could do. Another example of segmented lists occurs in the Scottish Commission for the Regulation of Care, whose inspectors are provided with a mobile search interface which allows them to search multiple databases simultaneously, with predictive query completion. As soon as three letters of the query have been typed, results matching this prefix from each database are displayed. Results are progressively refined as more letters of the query are typed.

15.4 Enterprise Search Evaluation

Enterprise search evaluation may be carried out for the purposes of scientific enquiry, for product testing, or for internal purposes of a company.

15.4.1 Published Test Collections for Enterprise Search

Big differences in information holdings from one organization to another and the fact that most enterprise information is company confidential pose extreme difficulties in constructing test collections on which search tools can be tuned and compared. Sometimes, the full set of data owned by a company becomes available outside the company due to bankruptcy. Enron is a well-known example and, although some researchers express ethical concerns about using it for science, many studies have in fact been based on the Enron email corpus.

Unfortunately, access to the full data set is only a partial solution to the problem of building a test collection. In addition, we need to understand what types of searches would be conducted during the normal operation of the company, what queries would be issued and what value would be placed on results returned. In general it may be difficult to contact the former employees of a bankrupt company and to persuade them to share information needs and judgements.

Even if we as scientists were granted permission to copy all of an organization's data and to study its employees' search behaviour – what they search for and which search results they value – how could we be sure that this organization is representative of others. How could we be confident that the best search engine on this company's data would also perform best on another company's completely different data?

The TREC Enterprise Track

To the best of our knowledge the only publicly available collections (corpus + queries + judgements) for enterprise search evaluation are those created by the TREC Enterprise Track. Because of previously mentioned issues, the Enterprise Track Corpora comprise only material published on external Web sites – A crawl of w3c.org, mailing lists from w3c.org (converted to Web pages), and the crawl of csiro.au mentioned in section 15.2. The interested reader is referred to that section, to the Track overview papers published on the TREC Web site[14] (*e.g.* [126, 439]), and to Chapter 4.

15.4.2 Internal Enterprise Search Evaluations

The main reasons why evaluation of the effectiveness of enterprise search tools is carried out in practice, are as follows:

1. R&D carried out by a search company to improve algorithms and to choose good default values for coefficients in the ranking function. Such R&D needs to be carried out on a wide range of test sets representing different enterprise retrieval environments.

2. Product comparisons leading to purchasing decisions.

3. Tuning of an existing system to make it perform better, within the context of a particular implementation. Such tuning may:

   (a) Cut costs by increasing the proportion of public enquiries which can be handled through the Web rather than requiring more expensive telephone or face-to-face support.

   (b) Increase employee productivity (*e.g.*, by avoiding effort spent recreating information which already exists) and company competitiveness.

   (c) Increase sales, by ensuring that potential customers can easily find product and service information and can find the most convenient way to buy (either on the Web or through traditional means.)

   (d) Improve the quality of decision making.

[14] http://trec.nist.gov/

(e) Reduce complaints.

Evaluation of enterprise search is no different in principle to evaluation of other types of search, though it is important to ensure that the evaluation faithfully models real enterprise search. The method of comparing two search facilities by presenting their results side-by-side in response to a single query (see section 4.5.2 and [1581]) has many advantages in this environment. Because the comparison tool replaces the search tool normally used by the people studied:

1. If the group studied is a uniform sample of the population using the search facility, the validity of inferences about that population may be subject only to unbiased sampling error. Sampling error can of course be reduced by increasing the sample size.

2. There is no need for experimenters to understand (or even know) what tasks are being performed by the searchers. All that need be recorded for each searcher is the vote cast for each search (*e.g.* "prefer A", "prefer B", "both useless", "both equally good".

3. Results sets are evaluated in their entirety. The evaluator may not like a results set in which there is substantial overlap in content or opinion even if most of the documents are highly relevant.

4. A person submitting a query knows why they submitted it and can evaluate the result sets obtained according to how well that set meets the need behind their query. Experimenters have no role in deciding: how many results should be judged, what grades of relevance should be available, what measures should be applied, or what penalty to apply to repetition within a results set. Instead, the searcher subconsciously applies whatever decision making process is appropriate in the context of their task.

5. Compared to most laboratory retrieval experiments with human subjects, side-by-side evaluations are conducted in real rather than simulated contexts. Furthermore, presenting A and B conditions simultaneously to the same person controls for differences between subjects and for differences in judgements made by the same person at different points in time.

Concern is sometimes expressed at the lack of sensitivity in side-by-side comparisons. No significant preference between systems A and B may be found even after dozens of searchers have evaluated reasonable numbers of queries, even though a TREC *ad hoc* evaluation may have found differences in MAP of a few percent. But is this really a disadvantage of the method? If System A is the one currently in production use, it is clear from the side-by-side study that any benefits obtained by replacing it with System B will be small. The volume of user complaints is unlikely to reduce.

Making n-way comparisons using the side-by-side tool is optimal for $n = 2$. With current display technology, it is possible to increase n beyond two but the number of judgement outcomes needs to increase and the potential to interfere with the work of the searcher through increased judging overhead increases. In [941] $n = 3$ and

subjects were asked to rate each panel on a scale, rather than expressing pairwise preferences.

An even bigger limitation of the side-by-side method is the inability to use it for tuning purposes. A typical enterprise ranking function will combine 20 or more variables. Tuning the combination function requires an amount of preference data which would be impractical to obtain through side-by-side comparisons. As on the Web, analysis of clickthrough data could be used for mining preference relations, see section 4.5.5.

15.4.3 Enterprise Search Tuning

For tuning an enterprise search system one can use a conventional but private-to-the-company test collection or take a machine learning approach and gather large quantities of data of the form, "For query Q, document D_1 is clearly preferred to document D_2" (based on frequency of clicks after controlling for forms of bias) [841, 844]. Judgments for a test collection, in the style of TREC ad hoc, may be made manually by employees of the organization or could rely on user click data.

Reliance on click data in evaluation is subject to risks. First, it is potentially subject to systematic bias in favour of documents whose title, URL and snippet in a result set tends to make them appear relevant when they are not, or vice-versa. Second, a search engine ranking function may exploit user click data in various ways, both query dependent and query independent. There is a risk that tuning the ranking using a cost function based on click frequency, will tend to upweight the click components of the function, leading to poorer-than-necessary performance on queries for which little click data is available or where the click data is misleading.

When using a test collection for tuning, the collection (documents, information needs, judgements and measures) should accurately represent the real situation. Otherwise, optimal parameter settings determined from the test collection might be far from optimal in actual use.

The workload of an enterprise search is faithfully recorded in the query logs maintained by the search engine. An obvious approach to unbiased evaluation is to take a uniform random sample from the query log and attempt to identify the most useful answers to the information needs assumed to lie behind the queries submitted. In [1389] it is demonstrated that the level of performance predicted for a search engine can vary substantially, depending upon how the query set is chosen. The ranking of a set of alternative search engines can easily change depending upon the set of queries chosen for evaluation.

A limitation of the workload sampling approach is the need to infer information needs from queries in the log. In some cases the interpretation is obvious, *e.g.* "pay scales", and "intellectual property policy", but in other cases, a little thought suggests multiple reasons why a particular query may have been submitted. In yet other cases, the meaning of the query is totally mysterious or the query is out of scope, *i.e.* that there is no identifiable useful answer within the document collection.

Another issue is that an enterprise search engine (particularly when used to provide external Web site search) is operated primarily for the benefit of the publisher rather than the consumer of the information. In many cases, the interests of the publisher and the site visitor coincide, but in some cases they do not. A publisher may want

to focus evaluation on the queries which are business critical to the organization. To illustrate, a bank may demand that its Web site search should give perfect answer sets to queries such as, "mortgage", "housing loan", "credit card application", but not be so interested in requests for past annual reports, or for fee schedules. A desire to bias search engine performance in order to achieve business or political goals is found in many organizations.

The open-source C-TEST Toolkit[15] [722] published by CSIRO provides a formal way of representing evaluation test files, which is capable of modelling many of the factors which are necessary for meaningful but reusable enterprise search evaluation:

1. Weights can be associated with each query in the test file, reflecting importance, be it determined by business factors or by frequency of submission.
2. Multiple interpretations for an individual query can be represented.
3. The fact that multiple documents in the collection are of equivalent value can be represented. This can prevent a retrieval system from gaining credit for returning multiple copies or versions of the same document.
4. Relevant answers can be assigned weights reflecting their contribution to meeting the information need behind a particular interpretation of a query.
5. The test file entry for a query can specify the appropriate judging depth given the need behind the query. For intelligence assessments or scientific studies, it may be reasonable to judge documents to rank 1,000. However, when finding the homepage of the HR department on the intranet, employees are unlikely to bother looking beyond the first ten results.

At the time of writing, maintainers of the open source retrieval systems Lemur, Terrier and Zettair have agreed to provide support for C-TEST formats.

15.4.4 What is it Reasonable to Expect?

The performance which it is reasonable to expect from an enterprise search facility lies somewhere on the continuum between the following extremes: the best possible answer at rank one, and a set of partial answers sprinkled among a lot of irrelevant results, as we now discuss.

The Best Possible Answer at Rank One

If we query a current Web search engine for the name of a company, such as 'Ford Motor Company', or the name of a mathematical concept, such as 'strongly connected component', then there is a high probability that the best answer page for the company (its homepage) or an authoritative definition of the concept will appear as the very first search result. This success relies on the richness of the Web search environment: link graph, anchortext, URL length and structure, user-behaviour data, etc. It also relies on the availability of information published specifically to answer these needs – company Web sites and Wikipedia or other sites created to provide high quality definitions and explanations.

[15] http://es.csiro.au/C-TEST/

A Set of Partial Answers Sprinkled among a Lot of Irrelevant Results

The task modeled by the TREC *ad hoc* (1992–1999) evaluation campaign (see Chapter 4 and [1654]) was essentially intelligence gathering from newspaper archives. When processed by state-of-the-art search tools over even quite small collections (*i.e.* half a million articles) TREC *ad hoc* queries, such as 'economic impact of recycling tyres' and 'dangers posed by the spread of fissionable materials from the states of the former Soviet Union', achieve much less satisfactory results. It was very typical that even the strongest TREC participants failed to find half of the relevant documents and returned five or more irrelevant documents in the first ten. In this type of search there are no definitive answers like www.ford.com, while signals that one article is more important than another are difficult to discern. There is no link structure, no anchortext, no site structure and, in TREC ad hoc, no user behaviour data. Above all, no single document exists which is designed to meet the information needs behind either of these queries. Furthermore, the query may not use the same words as documents it should match: *e.g.* 'states of the former Soviet Union' should match 'Russia', 'Ukraine', etc. (and possibly the Cyrillic equivalents of the same terms) while 'fissionable materials' should match 'U-235', 'Plutonium' etc. In this type of search environment, good search engines distinguish themselves from less effective ones by performing text-related operations better: query expansion (stemming, US-UK conflation, pseudo-relevance feedback, thesauri), document length normalization, and relevant passage upweighting.

Where on the Continuum does Enterprise Search Lie?

In a well-organised intranet which looks like a microcosm of the Web, the experience of searching for departments, people, and services can be at the happy end of the continuum. On the other hand, if the information to be searched consists of blocks of plain text in a database, or office documents dumped into an unstructured fileshare without metadata or a naming convention, baseline performance will be lower and users will be unhappy. In the latter scenario, a search effectiveness consultant might find ways in which current versions of reports can be distinguished from drafts, and a variety of query independent factors which could be used to improve ranking. At the same time, they might also suggest simple improvements to the way information is published and stored, as a means of improving search effectiveness.

15.5 Potential Reasons for Dissatisfaction

It has been previously noted that employee satisfaction with enterprise search and visitor satisfaction with Web site search are often low. Satisfaction depends upon "search and searchability", *i.e.* on both the effectiveness of the search technology, and how effectively information and services are published. It is sometimes the case that the best answer to a query does not match that query, perhaps due to language (for example, query is 'door', document discusses 'manually operated personnel egress mechanisms'[16]) or perhaps because the query words are only present in a graphic

[16]Thanks to *Private Eye* for this example.

(*e.g.* a scanned document). In these cases, it seems better to improve the way the information is published, rather than to attempt to modify the search technology.

All the current search technologies of which we are aware are statistically based. In these systems the retrieval score of a desired document with respect to a query depends upon a combination of the degree to which the query matches the document (and its annotations) and a static score which reflects the probability that this document is likely to be judged useful (as in Web search). See section 15.3.5 for possible components of enterprise static scores. The rank of the document naturally depends upon scores achieved by other documents. Tuning a ranking algorithm to the way information is published in a particular site can make a great difference to effectiveness.

Table 15.1 (page 673) lists several reasons why a desired document doesn't rank highly against a specific query. But problems of matching and ranking are only a subset of the problems which are encountered in enterprise search. A surprisingly high proportion of complaints about enterprise search tools actually arise because a wanted document is not even in the search engine's index! Table 15.2 lists several reasons why this may be the case. Note that you should be able to determine whether a wanted document is actually [visible] in the index, using highly specific queries. For example, search for the document's title as a phrase or search for words in the document's URL.

There are a number of specific matching issues which can cause ranking problems within organizational search. In many universities, *map* is a frequently submitted query. Often there is a set of matching pages with large numbers of incoming links and strongly matching anchortext which are not good answers to this query. These are "site maps" published by many Web sites within the university's domain. In similar vein, the query *president* (or *dean*) should retrieve the President's (Dean's) page ahead of the page for the Deputy President (Associate Dean) etc. Finally, the query *Bachelor of Engineering* should retrieve information for that specific degree ahead of combined degrees such as Bachelor of Engineering and Commerce.

There are a number of ways in which these matching issues may be addressed. Identifying them is left as an exercise for the reader.

15.6 Context and Personalization

Except when indexes are updated, a simple search engine delivers the same results for a query and presents them in the same way, whenever, and by whom, that query is presented. But in reality, not all users are the same and search performance may be improved if answers to the following questions can be obtained and exploited: Who is searching? What role are they playing? What are they interested in? Why are they searching? Where are they located? What task are they performing? What do they already know? What are they capable of understanding?

Personalized information retrieval represents only one aspect of a broad field of Personalization research. Pierrakos *et al.* [1262] survey this broad field and outline the possible functions of a personalization system. A fully personalized enterprise search system may provide a personalized portal layout, with user-specific information displays, customised look-and-feel, alerts targeted to the person, personal search history,

	Primary diagnostic	Specific Questions	Comment
1.	Does the document actually match the query?	• Are all the query words present in the indexed text of the document? • Are you implicitly relying on stemming, thesaurus expansion, partial matching, spelling correction, semantic understanding, language translation, or mind-reading?	Perhaps your search tool doesn't support these things.
2.	Does the document match the query text less well than other documents?	• How many occurrences of each of the query words does the document contain? • Do the query words appear in proximity to each other, particularly as a phrase? • Do the query words occur early in the document? • Are [all] query words present in the document title or main headings? (As a phrase?)	Many search engines assign scores based on features implicit in these questions. You may need to view the source of a document to be sure that query words do not occur in NOINDEX sections.
3.	Do shorter documents match the query as well as this one?		Many ranking algorithms assign content match scores which are normalized by document length.
4.	Do higher ranked documents receive more links or textual annotations which match the query?	• Do they have many links from other Web pages which use the query words in anchor text? • Have they been tagged many times with folksonomy tags matching the query? • Are many user clicks associated with the higher ranked document when this query is submitted?	These features are effective at improving the quality of Web search and can also be successful on intranets.
5.	Do higher ranked documents have features which may indicate that they are more popular or important?	• More incoming links? • More folksonomy tags? • More user clicks? • Shorter or simpler URL? • More recently updated?	Some search engines provide tools to allow display of these factors.
6.	Do higher ranked documents come from different repositories to the desired result?	• Does the ranking function "accidentally" favour results from particular repositories?	An administrator may configure a system to present results from one repository ahead of another, or encourage repository diversity.
7.	Is the target document very similar to another document appearing in the ranking?		The desired document may have been detected as a near duplicate and eliminated or pushed down from the ranking.

Table 15.1: A key document is in the index. Why isn't it top-ranked for this query?

	Diagnostic	Comment	Possible Remediation
1.	Does the document exist?	Surprisingly, this is a real cause of complaints. Maybe there should be a report on X, but actually there isn't.	Create missing content?
2.	Is it within scope?	This is a very common reason for failure. The location of the desired document (*e.g.* external Web site) is not included in the search scope.	Change default scope to broadest possible. Make sure scoping restrictions in force are stated.
3.	Am I allowed to see it?	It is typical for access to many documents to be restricted to certain employees. Are you logged in? The company may not want you to see this document.	If appropriate, initiate a change of permissions.
4.	Is it reachable by the gatherer?	If an intranet or Web site document is not linked to, it will not be crawled or indexed. With databases and fileshares, configuration errors may also result in missed content.	Ensure Web content is linked in. Check inclusion and exclusion patterns. Check mechanisms such as `robots.txt` and robots metatags.
5.	If this document is in a binary or proprietary format, such as JPEG, PDF or MS Word, is its text content able to be extracted?	• Is a filter for this type of content installed? • If a PDF, is text extraction permitted? • If text content is represented in graphical rather than text form, is the system set up to OCR the image? And if so, does the OCR software garble the query words?	Publish in accessible formats. Ensure necessary filters are installed.
6.	Did the document exist when the index was last updated?	Gathering, filtering and indexing operations may be carried out at intervals (*e.g.* weekly), rather than continuously. Documents published since the last gather operation will not be in the current index.	Check gathering logs. Rectify problems. Initiate regather.
7.	Is the document flagged to prevent its display?	• Has publication been suppressed by an administrator? • Does the document include NOINDEX tags or comments? • Has the document expired or passed its validity period?	Correct publishing errors.

Table 15.2: Why doesn't a wanted document seem to be in the search tool's index?

customised search scopes and, most relevantly to the present discussion, search results biased to the needs of the individual.

It is well established that search results for an individual search can be improved if its context is known and can be exploited. Teevan *et al.* [1571] quantified the potential for gain from personalizing search by asking 15 human subjects to rate 50 Web pages returned in response to a Web search query as *highly relevant*, *relevant*, or *irrelevant* to them personally, given an explicitly stated search intent. For queries supplied by the experimenters, they found a diversity of imputed intents. Even when the imputed intents were the same, subjects disagreed substantially in the ratings they gave to results.

Pitkow *et al.* [1275] demonstrated real search effectiveness gains in an experiment conducted with 48 Web users, grouped into novices and experienced Web users. They studied the value of inserting a client-side personalization system *Outride* between searchers and a Web search engine. Outride is a browser add-on which builds up a model of the user-based on their searching and browsing history plus their demographic and application use profile. It augments user-submitted queries and processes a very large set of results from the back-end search engine in the following ways: Results are categorised into "Have seen" and "Have not seen" and reranked with reference to a vector-space representation of the user's profile. Pitkow *et al.* observed dramatic reductions in both the time taken to complete a search task and in the number of user actions such as mouse clicks or keyboard entries.

The behaviour of a search tool can be customised according to characteristics of groups, individuals, or tasks. In discussing "usage-based" IR methods, Pitkow *et al.* [1275] point out that retrieval systems can work with different granularities of usage data and fall back to coarser levels as required. There is particular potential for contextualizing search by employees within an organization, because much more can be known about individuals, their roles in the organization, and the tasks that they are likely to be performing. However, the present author is a firm believer that a plain-vanilla option should always be provided and that searchers should always be able to find out what assumptions are being made about the characteristics of their search.

In discussing the topic of contextualized search, we first review the search engine controls and levers that are available to contextualise the set of results presented in response to a query, to improve its ranking (assuming results are to be ranked), and to optimise presentation of the set of results to the specific needs. Next we discuss the issues in client-side versus server-side contextualization. We go on to examine the potentially high dimensionality of a search context vector, how the dimensionality can be reduced in deriving a profile of search settings, how its values may be determined and how they may be communicated to a search server.

15.6.1 Controls and Levers for Contextualization

Assuming that something useful is known about the context of a search request, by what means can that context affect the behaviour of the search system? In general there are five categories of search engine controls: scoping, static ranking, query manipulation, dynamic ranking, and presentation. The settings of these controls for a particular individual or group may be recorded in a *search profile*. We will discuss

different types of profile and the ways in which they may be defined later in this section. Potentially there may be a complete global profile which effectively defines "plain vanilla". This global profile may be multiply overlaid in a particular search by partial profiles corresponding to groups, individuals and tasks, in the manner described by Pitkow *et al.* [1275].

Scoping

The scope of a search is the complete set of documents which may be matched against the query and which are eligible for presentation to the user. Scope is controlled by the repositories which are included in the search, by any exclusion filters which are applied to matching documents within those repositories, and by access restrictions applying to this particular search. It is easy to see that scope is a powerful tool for contextualizing search. For example, a technician in the R&D department may be more satisfied with search results if the CRM and finance repositories are not included in the search.

Exclusion filters can exclude documents from included repositories on the basis of file type, media type, genre, reading age, date, URL pattern, or metadata features. We have already discussed a form of personalized scoping, where documents are filtered from a results list because the person submitting the search request is not authorised to access them. Another familiar example from the Web is adult content filtering where knowledge of a person's preferences (or those of their parents) is used to scope search results. Although adult content is presumably rare in most enterprises, similar filtering techniques can be used to suppress certain types of documents (*e.g.* technical manuals) or certain intranet subsites (*e.g.* the employee childcare centre or the male employee hockey team) for certain individuals or groups.

Personal metasearch [1580, 1583] is a particular example of scoping in which an individual chooses the set of sources likely to be important to them over time. See Figure 15.5. For a particular query, the personal metasearch system may select a subset of the set of sources, on the basis of source characteristics, and/or the past behaviour of the individual.

Biasing

Rather than totally excluding a particular category of content by scoping, it may be more effective to bias the ranking algorithm against it. As we have seen, modern enterprise search engines include a static component in their ranking functions which is a weighted combination of many query-independent features such as: author popularity scores (*e.g.* HITS, OPIC or PageRank), user popularity scores (*e.g.* click derived measures), document recency, etc. Jeh and Widom [831] describe a system in which a global PageRank vector is augmented with personalized partial vectors to give a scalable implementation of personalized views of page importance/popularity on the Web. This scheme is relevant to some enterprise Webs but the scale of such Webs is very much smaller, to the extent that it may be feasible to maintain full personalized vectors, at least for groups of users within an organization. An obvious form of contextualization of user popularity data is to record user interaction data on an individual or group basis and to use popularity scores specific to the individual or to a group to which they belong.

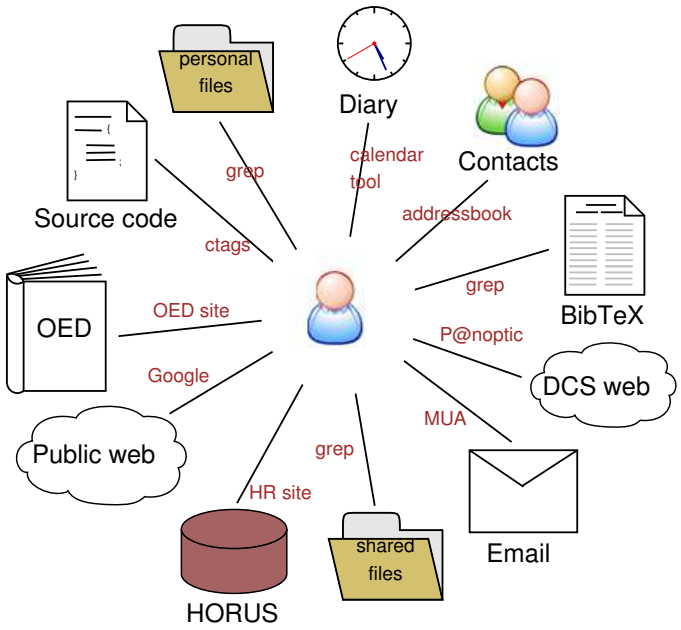

Figure 15.5: A particular personal metasearch configuration, showing the set of information sources included in the unified search. Note the presence of sources at the personal, workgroup, enterprise and external levels. Diagram courtesy of Paul Thomas, with permission [1581].

All of the features which can be used for scoping may also be used to bias search results for or against the feature. For example, instead of excluding technical manuals, the ranking function may be biased against them. If there are few relevant documents of other types, technical manuals can be retrieved. In further illustration, documents authored by the managing director may be upweighted or downweighted. In short, the behaviour of the search system can be contextualized using different vectors of query-independent weights, depending upon context.

Manipulating Queries

The query submitted by a searcher may be manipulated in various ways in order to contextualise results. In a simple case, group-specific thesauri may be used to interpret ambiguous terms in ways most appropriate to the group. For example, the query *CRM* may be augmented with *customer relationship management* for the sales department and *carbon reinforced mouldings* for the production department.

An extra degree of automation and sophistication is achieved by expanding the query using a variant of the technique of pseudo-relevance feedback, as described by Teevan *et al.* [1570]. These authors expanded a user's Web search query using pseudo relevance feedback on a search of documents on the user's personal computer. A larger than usual set of results was requested from the Web search engine and it was reranked

by locally re-scoring the documents using the expanded query. Teevan *et al.* observed a small but statistically significant improvement over the baseline ranking.

Manipulating the Dynamic Ranking Function

As seen in Chapter 4, almost all functions for scoring the relevance of document content to a query are parametrized. There is therefore potential for adjusting parameters to improve result quality for particular users or groups of users, although it is not intuitively clear why particular parameter settings might suit some users better than others. There is perhaps more scope for tailoring the use of textual annotations to the benefit of particular groups. For example, looking only at folksonomy tags applied by people similar to the person, or taking account only of click-associated queries submitted by a group to which the person belongs.

Particular linguistic features or transformations may potentially suit some users (or tasks) better than others. For example, people may prefer heavy stemming, light stemming, no stemming, or stemming targeted at a particular language. When performing certain tasks, it may potentially be advantageous for the search engine to automatically conflate US and UK English spellings. Some people may prefer that a query comprising unaccented letters should match accented versions of the same words, *e.g.* 'canon' should match both 'canon' and 'cañon', and others may not.

Some ranking functions include components designed to improve the diversity of results returned. For example, Carbonell and Goldstein [332] make use of *maximal marginal relevance*. In their scheme a document's position in the final ranking depends upon a combination of its similarity to the query and its dissimilarity to the vector-space centroid of documents appearing above it in the ranking. Other schemes try to include results from a diversity of sources, or a diversity of result types. The definition of diversity and the rewards applied to it are configurable on an individual or group basis in certain retrieval systems, but we are unable to cite examples where this is used to advantage in practice.

Customization of Human Interfaces

There are many ways in which the presentation of search results may be customised to the needs of individuals or groups. Do you like to see a ranked list of results or a grid of thumbnails? Where does your preference lie on the spectrum of many terse results on a search page versus a few detailed ones? Do you have preferences regarding colours or fonts? What language do you prefer? If you could be shown a summary of all the results rather than a list or grid of individual results, would you choose it? Do you like to see facet counts, query suggestions, etc. and other add-ons? Do you like to have news items relating to your query highlighted at the top of the search? Do you prefer or need to have search results spoken to you rather than displayed on a screen?

The above examples apply to the presentation of results, but customizations are possible on the input side too. Do you prefer to type your query, or speak, or write it? Do you like the query input to auto-complete for you? – Based on your own query history, or on the shared histories of a group of which you are a member? Many of these issues relate to accessibility, which can be critical in an enterprise context. If

an employee with a disability is unable to use the organization's search facilities, or only with difficulty, they may struggle to be productive.

Finally, employees may access enterprise search facilities via telephones or mobile devices with limited screen area. Ideally the limitations and capabilities of these devices can be recognised by the search facility and the interaction structured appropriately. On the Web, mobile devices fitted with GPS systems, or indeed the IP address of a user's workstation can be used for *localization*, the tailoring of search results (and the type of interaction) based on the user's country or region. Some multi-national companies may be able to exploit these capabilities internally, but the vast majority are too small or too localized to do so.

15.6.2 Contextualization: Local, Enterprise or Global?

Most of the preceding discussion of contextualization has been deliberately vague about where the customization of search might occur. It is now common for individual employees playing knowledge or information rich roles in an organization to be allocated a personal computer (PC) for their own use. Sometimes that PC is set up to work in a centralized way, in which software is accessed from centralized enterprise or workgroup servers, and files and email are stored there. However, in many cases the PC must operate in a standalone fashion, either because of company IT policy or because it is a laptop which can be used at home, on customer sites or at meetings and conferences. Whatever the detailed arrangements, PCs are able to collect vast quantities of personal interaction data – what the person received, downloaded, viewed, filed, sent, edited, printed, bookmarked and searched for. The PC can potentially monitor external activities too – what social networking sites the person accessed and interacted with, what "tweets[17]" and instant messages they sent or received.

Almost all of the information which might be useful for personalization is primarily collected or collectable at the person's PC, although some of it is also collectable by servers in the organization and a lesser amount is collectable by search and other services (such as ISPs) external to the organization. A small but increasing exception to this generalization is that a great deal of communication and information interaction now tends to occur on mobile devices. Some executives process most of their email via devices such as Blackberries and iPhones. However, email and contacts are usually synchronised with their PC.

An important element of contextualization is the nature of the task being performed (see section 15.2 for some examples). There is great potential to exploit this within an enterprise, because the task may be accurately inferred from the application currently being run. For example, if an employee performs a search while running their company's project cost estimation application, it would be reasonable to infer that their search was conducted in the context of estimating project costs. That inference would be even more reliable if the search function were embedded in the application.

[17] http://twitter.com

Client or Server?

As noted above, the bulk of information useful for personalization and other forms of contextualization is typically available on a person's PC. A client PC is clearly the best place to maintain personal and task profiles for search. If profiles are kept only on a secure PC, the privacy risks discussed above are controlled. Unfortunately, part of the task of personalizing searches conducted on external search engines (such as Web search engines) is best done at the search engine, not at the client PC. This is where modifications to scope, static scores, and ranking can be fully effective.

In their survey of Web Usage Mining as an aid to personalization, Pierrakos *et al.* [1262] outline a wide range of methods which can be used to collect Web usage data, ranging from logging toolbars at the client machine, to packet sniffers, Web logs maintained at various points between client and server, and server logs. They also outline the problems with reliable identification of individuals and session boundaries in remotely collected data.

Web search engines keep profile information for individuals, maintained from one query to the next by means of cookies. However, the profile so maintained is incomplete, particularly if the individual uses multiple search engines and does not share email or documents with the search engine. Also, the profile specifies only the personalization supported by the particular search engine. On the other hand, search facilities operated by and within organizations, including personal metasearch, offer more potential scope for customization, and privacy issues are more easily managed.

Of course, it is possible to communicate interaction histories, profiles or part profiles to an external search engine, but we are then beset by a number of issues: Our desire to maintain privacy is thwarted; and we must work out how to communicate the useful parts of the profile to the search engine in a form which achieves the desired effect, without generating excessive network traffic or server load.

15.6.3 Privacy of Profiles

There is good reason to maintain the privacy of interaction histories and personal search profiles. The material could provide very valuable information to a competitor, very useful data for targeting advertisements, and, in less likely scenarios, compelling material for divorce lawyers, blackmailers, police, and foreign intelligence services. Don't forget that some of the profiles we are discussing belong to people whose interests and activities are of more critical importance than those of ordinary people. Any information about the searches conducted and the documents read by a wealthy corporate raider or by senior executives in the US Federal Reserve would presumably be of great interest to stock market speculators. Other groups of people may find greater value in the search profiles of the president of a country or the head of a narcotics agency!

In 1997, a proposal was submitted to the World Wide Web Consortium for an Open Profiling Standard[18] which would enable secure exchange of profile information, but this has not yet been adopted.

[18] http://www.w3.org/TR/NOTE-OPS-FrameWork

15.6.4 Defining, Creating and Maintaining a Profile

There are some researchers who are interested in the total context of search behaviour and who would consider all of "my life bits"[19] as relevant context, along with a full description of meteorological, geographical, psychological, and social factors. In other words the objective is to understand humans and their online behaviour rather than to improve the value of an individual search "transaction".

Restricting ourselves to the more prosaic, practical dimension, it is very difficult to specify precisely what information should be recorded in order to derive settings of the controls and levers outlined above which are optimum for a particular search by a particular person. The types of interaction information mentioned above, which may be useful for generating a search profile, can be recorded by software run at the local operating system level or as add-ons or plugins to browsers and other personal or enterprise applications. It is easy to keep a full record of all interaction events (and objects) and the times at which they occurred [17, 1570]. Other less complete records can be kept at greater distance from the user, in applications such as proxy servers or search engines.

Another question to be answered is how to determine which profile should be applied to a search. The same person may prefer different profiles (or none) depending upon the activity they are engaged in at a particular time. Can we build an automatic system for determining the right profile which achieves 100% accuracy? Likely not. But otherwise how do we explain to a user, without confusing them, what profiles are available and how they differ. In an enterprise the best approach may sometimes be to have the user select (if they wish to) among a small set of obviously named group profiles, such as sales, finance, HR, R&D, and plain vanilla.

15.6.5 User Modeling

The process of automatically generating profiles may be described as *user modeling*. A number of different types of user models have been described.

Ontology Vectors

Pretschner and Gauch [1302] describe a system in which a user's profile consists of a vector of weights for each of 4,400 hierarchical categories in a published ontology. Each category is represented by a document vector representing the amalgam of ten exemplar documents for the category. When the user visits a Web page, that page's similarity to each of the categories is computed and their profile weights are updated as a function of these similarities, the viewing time for the page and the length of the page. This research found that profiles converged over time and that the profiles could be used to rerank and filter results from a back-end search system. Modest but worthwhile gains on 11-point average precision were obtained through reranking while filtering results were more equivocal.

As described earlier in this section, Pitkow *et al.* [1275] achieved substantially greater gains, albeit with a quite different evaluation methodology. They used a similar ontology vector profile though they do not give precise details. Their method used query augmentation as well as reranking.

[19] http://research.microsoft.com/en-us/projects/mylifebits/

Relevance Feedback Methods

Teevan *et al.* [1570] describe methods for reranking top-50 search engine results based on a range of different types of user profiles. A simple method producing substantial gains is to promote in the ranking URLs from domains recently visited by the user. A much more sophisticated model represents the user by the index of their desktop search tool, *i.e.* by all the files, Web pages and email messages on their personal machine. The desktop search index is used as a pseudo-relevance feedback engine which generates an expanded query used to rerank the 50 results from the Web search engine. Unfortunately, both the original Web ranking and the URL reranking outperformed this highly personalized reranking. However, a mixed method combining both the original Web ranking and the personalized ranking was found to improve on the raw Web ranking slightly but significantly.

User profiles studied by Waern [1662] consisted of long lists of terms, either manually constructed by the user or automatically derived. The study found that users were generally unable to improve on machine learned profiles but pointed out that user involvement in profile maintenance was essential to enable correction of errors made by an automatic profile generator.

Characterizing Users by their Clicks

We have already discussed the work of Joachims [841] on making use of clicks to learn better ranking functions. Joachims mentioned the potential use of clicks for personalization but did not report results. Using Microsoft search engine logs, Dou *et al.* [508] performed a large scale evaluation of five personalized search strategies, of which two were based on clicks and the others based on automatically derived profiles. They found that personalization has the potential to significantly improve search quality but that the benefit varied considerably from query to query, sometimes even causing harm. They found that queries with high click entropy are the ones which benefit most from personalization and that simple click-based personalization is consistently beneficial while profiles which attempt to capture user interests are less stable. The method based entirely on a user's past clicks is only capable of improving queries which this user has previously submitted. To address this limitation, another method used click patterns to assign users into groups with common interests, and the group click history was used for personalization. However, the results for personal and group click profiles were indistinguishable.

Language Models

Tan *et al.* [1556] describe the extension of the language model framework of information retrieval to include both short-term (current session) and longer-term historical language models derived from click behaviour. Such historical models can be considered another form of profile.

Biasing PageRank

A quite different form of user profile is the personalized PageRank vector model proposed by Jeh and Widom [831], which has been previously discussed in section 15.6.1.

15.6.6 Implicit Measures

Kelly and Teevan [896] provide an overview of the use of implicit measures derived from user behaviours in retrieval, filtering and recommendation. Table 1 in their paper lists five classes of user behaviour – Examine, Retain, Reference, Annotate and Create – which may be observed and used, for example, in profile building. Table 1 also identifies the minimum scope of the items being acted on by specific behaviours in the classes, while their Table 2 assigns a considerable number of prior studies into the Table 1 grid.

Researchers at Microsoft have extensively studied the use of implicit measures (obtained from instrumented versions of their browser) in improving the quality of Web search results. Fox *et al.* [580] established that a probabilistic combination of implicit measures such as clicks and page dwell times could accurately predict explicit judgements made by users. Agichtein *et al.* [18] extended this work to include query-dependent measures and proposed a distributional model robust to noise. Agichtein *et al.* [17] showed that implicit measures, when combined in a large scale machine learning framework, can be used to improve Web search performance, either by reranking the original result set, or by integration into the base ranking function. Their study involved 3,000 queries and 12 million user interactions. None of these studies make use of implicit measures for purposes of personalization but there is clearly potential for them to be used in constructing individual or group profiles. Given the differences between one organization and another and the sparsity of interaction data, it is not clear to what extent the lessons of this work can be applied in the enterprise.

White *et al.* [1687] point out that while explicit relevance feedback can be beneficial, it imposes a load on searchers. They analyse an implicit version of relevance feedback (IRF) in which user actions such as reading, scrolling and saving are used to infer relevance judgements. Although IRF is less likely to be beneficial, they report that users, particularly novices, preferred it. They also found that IRF is more valued for complex search tasks and more likely to be used in the middle stages of search activity than at either the beginning or the end.

15.6.7 Information Filtering

Personalizing search results can be seen as a bringing together of tools from Information Retrieval (IR) and from Information Filtering (IF). Hanani *et al.* [696] provide a detailed conceptual framework for IR and IF and compare and contrast the two. Generic search results are produced and then filtered to remove items in which the particular user is unlikely to be interested. Personalization aims to achieve better results from *ad hoc* searches by combining the terse specification of an immediate need (the query) with a more expansive, longer-term profile. In the case of routing and alerting systems, we still see a combination of IR and IF techniques but in this case there is no immediate query. Instead, a long-term profile is registered with a search service. Newly created or discovered documents are matched against this profile and if the match is good enough the document is forwarded to the user by email or RSS feed.

For decades, organizations like Lexis-Nexis have offered selective dissemination of information (SDI) services in which users register a profile consisting of a Boolean query and are sent all documents which match the filter. In this model the user takes

on the responsibility for creating the filter query and must maintain it to ensure that they do not either miss important documents or pay for documents which are not really of interest. More recently, Google has provided a similar alerts facility in their Web search engine. Documents which newly arrive among the top ranked results for a user-registered standing query are candidates for being sent to the user as an alert. Google researchers Yang and Jeh [1740] discuss perceived problems with this alerting service and describe and evaluate methods for automatically extracting alert profiles from a user's search history. The challenges are to identify long standing interests in the query log for which the user is likely to be interesting in seeing new documents.

An alternative approach to Information Filtering is to automatically associate an individual with a group and to use a group profile to customise results. This is called Collaborative Filtering and (CF) systems which use it are sometimes known as Social Recommender Systems.

15.6.8 Social Recommender Systems

Modern search engines, both on the Web and in the enterprise, perform a type of generic collaborative filtering in their base ranking methods. Documents which are linked to by many authors, or which are tagged or clicked on by many readers, tend to receive higher static scores and to appear higher in result rankings. Individual searchers thus benefit from the wisdom of the populations of authors, browsers and searchers. Resnick and Varian [1343] describe a number of recommender systems and how they work. To achieve personalization within a CF framework, one can identify groups within the overall populations and associate individuals with appropriate groups.

Heer and Chi [740] studied methods for categorising user sessions on the xerox.com Web site and performed a user study in which users were asked to perform realistic information finding tasks. By using a combination of features such as browsing path and page dwell time they were able to achieve very high clustering accuracy. It is not clear how quickly a new visitor or a new browsing session could be classified and whether the classification could be used to improve search.

An online shopping site can effectively increase its business by drawing a customer's attention to items that they are likely to be interested in. "People who bought item X also bought item Y." This problem can be addressed using information retrieval methods, by treating an item selected by a customer (or the accumulated list of purchased items) as a query and retrieving related items. However, as noted by Linden *et al.* [1036], Amazon found that search-based methods failed when users make large numbers of purchases. Instead they use a related item method, with the vast item-item similarity matrix computed offline. Two items are considered closely related if they both tend to be purchased by the same customers.

The interested reader is referred to Adomavicius and Tuzhilin [15] for a comprehensive review of content-based, collaborative and hybrid filtering methods.

15.7 Trends and Research Issues

The challenge of enterprise search is to provide knowledge-intensive organizations with a single-query search interface to all their document content. The ideal enterprise

search tool will provide results of sufficient quality to support additional functionality, such as business intelligence analysis, profile building for law enforcement, knowledge mining, report generation, and multi-document summarization. Note that an enterprise search tool is a natural platform for these functions because it brings together documents and data from multiple repositories and converts them to compatible and accessible formats. In recent years commercial search products have moved to support or integrate more closely with business applications.

However, as noted in the introduction to this chapter, there is a wealth of evidence to suggest that search facilities within enterprises have generally not yet approached the level of user satisfaction achieved by current Web search engines. Relatively few employees have access to a search tool which spans the enterprise's information resources yet routinely returns the most useful results in response to queries. This seems surprising given the very large productivity and competitiveness benefits likely to flow from highly effective enterprise search. A major cause is the lack of research into the specific problems of enterprise search, in turn caused by the lack of appropriate enterprise search test collections, in turn caused by the confidentiality of corporate documents and information needs and the large variation across organizations. There is a slow trend toward development of test collections which at least cover parts of the enterprise search space. Hopefully, problems of confidentiality can be soon overcome and the pace of research accelerated.

In the meantime, research continues in a number of areas important in enterprise search. First, distributed information retrieval, specifically personal metasearch, which relates to the problem of the federation of heterogeneous information sources within an enterprise. Second, the ability to find effective analogues of Web search ranking factors, "behind the firewall". Third, personalization, customization, and support for diversity in search results. Fourth, linguistic functions such as synonym detection, entity extraction, translation, summarization, and query suggestion.

Meeting the challenge of enterprise search requires contributions from all components in the system, including the initial creation and publication processes. Progress continues to be made, but more is needed in engineering, research, adoption of standards, and commercial practice.

15.8 Bibliographic Discussion

General information retrieval problems covered in other chapters, such as crawling, indexing, ranking, result presentation, summarization, and multimedia retrieval are all important within the domain of enterprise search. The reader is referred to the relevant chapters. The present chapter has mainly focused on the particular issues which give enterprise search its distinctive character.

First, the many engineering challenges [9, 274, 675, 719, 1535] which must be solved within an organization in order to obtain a quality corpus of text documents for indexing: adapting to the set of repositories and applications deployed by that organization, in order to extract documents; efficiently scanning shared file systems for files containing text; accurately and efficiently extracting text from within binary files, such as PDFs and office documents; authenticating against the security systems in force. In some cases, extracting documents from a repository is not feasible and

the search capability of that repository must be federated with the main enterprise search tool.

Second, the problems of ranking documents and presenting results within heterogeneous collections, where valuable sources of ranking evidence, such as links and anchortext, may be present in some sub-collections and absent in others. To the best of our knowledge very little work has been done on optimising single-index retrieval from a very heterogeneous collection. Even in the extensive literature on distributed information retrieval, very little apart from the PhD work of Thomas [1580, 1583] and the Stuff I've Seen system [518] relates to federation of genuinely heterogeneous repositories. It is unclear whether conclusions drawn from experiments in which sources are simulated by artificially partitioning the TREC Ad Hoc collection have any applicability in the enterprise federation context. Third, when presenting results it is typically the case that an initial ranking must be filtered [127] to remove documents which the particular searcher is not entitled to see.

Fourth, the problem of evaluating enterprise search is made difficult by the confidentiality of documents and information needs and by the huge diversity between organizations as far as quantity of information, number of repositories, number of document types, and nature of searches conducted. This makes it difficult to study enterprise search and difficult to tune enterprise search systems "in the factory". Hansen and Järvelin [697], Freund *et al.* [589, 590], and Hertzum and Pejtersen [756] have studied real enterprise search, while Craswell *et al.* [439], Bailey *et al.* [126] describe test collections oriented toward enterprise search and expertise finding. The reader is referred to TREC Enterprise Track overviews and participant reports for TREC 2005 – 2008. These are available online at http://trec.nist.gov/proceedings/proceedings.html.

Finally, Grefenstette's keynote presentation at ECIR'09 [675] outlines eleven specific differences between Web and Enterprise Search.

The important topics of personalization and customization are not specific to enterprise search but have important potential and particular features in this space. In addition to the papers cited in the section above, particularly review articles such as [15, 896, 1262], a good place to start reading about contextualization and personalization is the proceedings of the "Information Interaction in Context" workshops.[20]

Two specialist enterprise topics have very high economic importance and deserve particular mention. Legal discovery searches over company records have the potential for very high impact. Useful overviews are provided by Roitblat [1378] and Baron *et al.* [149]. Patent retrieval is another legally oriented task of vital importance to many major companies. Patent retrieval has been studied within the NTCIR series of workshops organised by the Japanese National Institute of Informatics, starting with the third workshop in 2002. See the online proceedings at http://research.nii.ac.jp/ntcir/publication1-en.html. Recently, the Information Retrieval Facility (IRF, http://www.ir-facility.org/the_irf) in Vienna has been established with a goal to promote and support open information retrieval research with a particular focus on patent retrieval. It provides data collections, and large-scale computing infrastructure to support research projects and has sponsored the Intellectual Property track in CLEF-09 and the Chemical Retrieval track in TREC-09.

The broader topic of enterprise information architecture is addressed in the 2006 edition of Morville and Rosenfeld's book, *Information Architecture for the World*

[20] http://irsg.bcs.org/iiix2008/

Wide Web [1157]. A report reviewing currently available enterprise search options, *The Search and Information Access Report*, is published at intervals by CMS Watch.[21] Other reviews available are targeted at the corporate sector.

[21] http://www.cmswatch.com/Search/Report