Beyond Document Similarity: Understanding Value-Based Search and Browsing Technologies

Andreas Paepcke†[1]
Hector Garcia-Molina†
Gerard Rodriguez-Mula††
Junghoo Cho†
†Stanford University (paepcke, hector, cho@cs.stanford.edu)
††Universitat Politecnica de Catalunya(gerard@ac.upc.es)

Abstract

In the face of small, one or two word queries, high volumes of diverse documents on the Web are overwhelming search and ranking technologies that are based on document similarity measures. The increase of multimedia data within documents sharply exacerbates the shortcomings of these approaches. Recently, research prototypes and commercial experiments have added techniques that augment similarity-based search and ranking. These techniques rely on judgments about the 'value' of documents. Judgments are obtained directly from users, are derived by conjecture based on observations of user behavior, or are surmised from analyses of documents and collections. All these systems have been pursued independently, and no common understanding of the underlying processes has been presented. We survey existing value-based approaches, develop a reference architecture that helps compare the approaches, and categorize the constituent algorithms. We explain the options for collecting value metadata, and for using that metadata to improve search, ranking of results, and the enhancement of information browsing. Based on our survey and analysis, we then point to several open problems.

Categories and Subject Descriptors: Computer Systems Organization, Information Systems.
General Terms: Design, performance, management.
Keywords and phrases: Information retrieval, information filters, metadata, relevance, World-Wide Web, search engines, ranking, links, hypertext, collaborative filtering.

---

1. Andreas Paepcke; Stanford University; Gates Computer Science, rm 426; Stanford, CA 94305; Phone: 650-723-9684. Fax: 650-725-2588

This material is based upon work supported by the National Science Foundation under Cooperative Agreement IRI-9411306. Funding for this cooperative agreement is also provided by DARPA, NASA, and the industrial partners of the Stanford Digital Libraries Project. Any opinions, finding, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the other sponsors.

1 Introduction

With the advent of the Web, information of broad interest has rapidly moved online. The penetration of online information into the everyday lives of broad population sectors has revolutionized the way we approach information-related tasks. Both the production and consumption of information is increasingly in the hands of nonspecialists who have generated a dazzling palette of resources.

Unfortunately, this very exciting development also produces the widely suffered problem of the Web: information glut that often prevents the discovery of useful information, while overwhelming our senses and time reserves when thrust upon our screens.

The arsenal of defenses against this problem has mostly been stocked with techniques from the field of information retrieval (IR). In contrast to database systems, IR systems were ready for dealing with the relative poverty of structure prevalent on the Web. IR systems have always been geared towards unstructured text, while databases specialized on structured information.

Most of the information retrieval systems common today perform searches by computing some similarity measure between a given query and the content of a collection. These systems differ from each other mainly in how effectively they compute that similarity. While sophisticated similarity-based techniques have fueled important progress for Web search engines, they are increasingly being overwhelmed by the amount of information they are confronting. Apart from sheer volume, one particularly vexing problem is that IR techniques only deal with text. More and more often, important information is contained in applets or audio and video clips, or text is embedded in graphics, and is therefore difficult to access. IR techniques fail in those cases.

Fortunately, there are emerging solutions that help address these issues. Contemporary research has begun to supplement basic IR approaches with techniques that collect indicators of 'information value', which are independent of similarity with any given query. These techniques then exploit this value metainformation to help users throttle the flow of information by passing only 'valuable' items. We collectively call these approaches 'value filtering'. Since value information can be attached to non-textual elements, value filtering has an important role to play in accessing previously non-searchable information.

Value filtering is related to the notion of 'information relevance'. The definition of relevance, the criteria that influence human relevance judgment, as well as appropriate measures of relevance have seen several decades of investigation and discussion (for surveys, see [1, 2]). As we will see, value filtering techniques attempt to use relevance judgments, but also query-independent methods for improving the subjective quality of retrieved information.

This survey focuses on recent technical approaches towards this goal, particularly in the context of the World-Wide Web.

Success in developing value filtering solutions is crucial if information systems like the World-Wide Web are not to collapse under their own weight. With the proliferation of new media on the Web, and the immense pressure of information growth bearing down on search and filtering facilities, an understanding of emerging alternatives and adjuncts to traditional similarity search is an urgent need for developers.

Collections of articles on existing experimental prototypes (e.g. [3, 4]) have been compiled in the past, but they do not explain how the various techniques are related, and how they might be combined and extended.

To address these shortcomings, we survey and classify different approaches to value-based information filtering. We present a conceptual architecture that enables an understanding of all the emerging techniques at the same time. Using this unified view, we explain these techniques, and illustrate their realizations in example systems that have been reported in the literature, or are known to the authors. It is not our intention to include a comprehensive list of example systems, but only to survey the different categories of techniques used.

Early examples of value filtering rely on explicit user participation for generating the value metainformation; users are asked to evaluate the information they retrieve, and the search machinery takes these opinions into account during future searches. Recently, techniques for automatically extracting value information about documents or collections have gained ground. Some analyze the structure of a collection, such as the linkage between World-Wide Web documents. Given two documents, they might favor the one with more incoming links. Other techniques observe how users access the collection, and draw value conclusions from these observations. For example, frequently accessed documents might be assumed to be more valuable. Another set of techniques attempts to maintain user profiles that reflect users'

interests. When deciding among two documents, systems that use these techniques will favor the document that most closely matches the current user's profile.

Value filters are used for several purposes. The most prevalent is to cull documents from result sets generated through standard IR techniques. Filters are also used to guide users while they browse. For example, when presenting a Web page to a user, a browser might highlight selected links that lead to other pages deemed particularly valuable. A third use of value filtering is selective dissemination of information (SDI). SDI systems continuously scan collections or information streams for 'valuable' information which they distribute to appropriate users.

But value filtering not only enhances a system's effectiveness by increasing precision. Another important use is to improve search system response time. For example, highly valuable information can be made more cheaply accessible through special indexing.

Value filtering systems thus differ from each other in their choice of value metainformation they extract, in how they go about the extraction, and in how they then use the metainformation to improve the overall system. We partition the techniques roughly into two categories: Content-based, and action-based approaches. Some of the systems we use to illustrate these approaches employ more than one technique and thus fall into multiple categories. Thus, we sometimes cite a single project in multiple places.

For the purpose of effective discussion in the following sections, Figure 1.1 shows a conceptual architecture that contains the basic elements of value filtering systems. Any

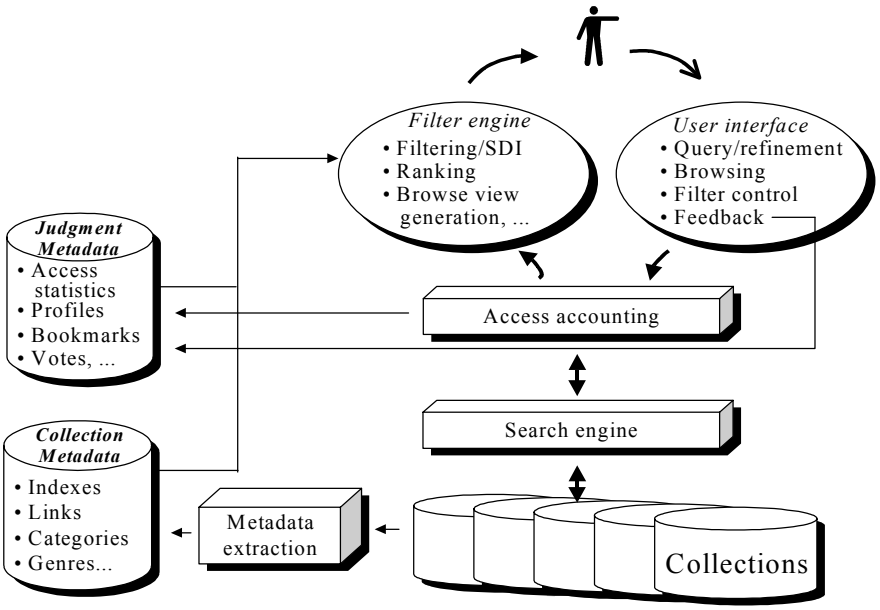

Figure 1.1: Conceptual architecture of value filter systems

given filtering system generally does not contain all the elements shown in Figure 1.1, and most systems are not explicitly organized according to this reference architecture. We will use this architecture to help relate the different systems to each other, and to clarify the different techniques they use.

Section 2 will explain in detail how each element of Figure 1.1 functions. Briefly, collections (bottom of figure) might be the Web, one particular site on the Web, a set of databases, email folders, or information streams, such as a news feed. The thick arrows indicate the information flow to and from users. Users submit and refine queries, or they use a browser to scan through a collection (right oval near top). A search engine extracts potentially relevant information, and passes it to a filter engine (left oval near top). Using value information, the filter engine removes some documents, ranks the remaining ones, or generates enhanced browser views which are passed up to the user. Filter engines thus not simply remove records from the information stream. They may also annotate, organize, or otherwise enhance the information that is eventually presented to the user.

Figure 1.1 shows two repositories of metadata along its left edge. These repositories hold the information needed by the filter engine. The collection metadata is maintained by the metadata extraction module (bottom). This metadata reflects

value information that can be gleaned from analyzing the collections. The judgment metadata repository records directly or indirectly obtained document relevance feedback, which might be provided through the user interface. It also contains user profiles, and access statistics that are generated by an access accounting module, which is interposed into the information stream.

The following two sections explore existing content and action based value filtering techniques. Section 4 outlines how value filtering is used, and points to several open issues.

2 Content-based Value Filtering

When a filtering technique relies on information that is contained entirely within the source collection it feeds on, we call the technique 'content-based'. Conceptually, referring to Figure 1.1, content-based techniques use a variety of metadata extractors to collect or deduce metadata about documents and collections. Filter engines use that metadata to filter or rank search results, or to create enhanced browser pages.

Content-based techniques fall into four sub-categories that emerge when examining how the techniques extract the metadata they use for filtering. Some techniques analyze individual documents, while others extract from entire collections. Techniques in the third category try to deduce value from the context in which information is embedded, while the forth category relies on manually placed tags within documents. In the following sub-sections, we look at each of these sub-categories in turn.

2.1 Document Analysis

Metadata extractors in this sub-category derive metadata by analyzing the individual documents in a collection. Sometimes the analysis is based purely on examining words contained in a document. One example is PHOAKS [5]. This system examines large numbers of Usenet messages and finds URLs within them. When it detects a URL, it attempts to deduce whether the URL is intended to be a recommendation. For example, PHOAKS checks which words surround the URL. If the words seem to be related to advertising, the URL is not categorized as a recommendation. The system thereby collects a set of URLs it believes to have been recommended.

Notice that PHOAKS does not necessarily operate in the context of a particular query. In contrast, TileBars [6] is a technique that also relies on document analysis, but is intended to support users in manually filtering result sets returned in response to queries. TileBar systems maintain collection metadata about the location of words within a document. The system's filter engine then uses this metadata to provide visual clues about where in a document, and how often, query terms occur. Users viewing these enhanced displays deduce value information from the keyword positions by using knowledge about traditions of document structure. For example, keywords occurring in the center of an article are expected to be part of the document's central theme. Keywords occurring only at the end, might be pointers to future work, or tangential closing thoughts.

Another kind of content-extracted metadata is vocabulary complexity. Some systems [7] attempt to rate the reading level of documents by analyzing how many 'difficult' words occur in them. This information can then be used to filter out documents that are too difficult, or that might be too simplistic for any given reader.

Some filters attempt to determine the 'genre' of the documents they encounter. Examples for document genres are newspaper articles, scholarly journal articles, interviews, advertisements, and press releases. Genre membership is assumed to help in assigning value to documents: maybe for some tasks, a scholarly article is considered of more use than an advertisement.
Various document analysis techniques are used to help metadata extractors determine genre. Extractors can base analysis purely on examining surface features within a document, as was argued in [7]. Another experimental approach attempts to derive strong genre predictors automatically through machine learning [8]: After being presented with a number of certified samples of one genre, such as citations of books, this system's metadata extractor first searches the Web for other examples of that genre, and then attempts to derive patterns that predict the genre.

Rather than analyzing the contents of documents, some approaches look at structure for clues about value. For example, the length of a document might be used to draw conclusions about its value. Similarly, when evaluating the intented use of a URL in a Usenet message, PHOAKS considers the URL's position within the document. If the URL is part of the sender's signature, it is assumed to be a pointer to the sender's home page, rather than a recommendation. If the URL is part of a quoted section of the message, it is also not counted as a recommendation.

2.2 Collection Analysis

Increasingly, filtering systems attempt to analyze not the content or structure of individual documents, but the structure of collections. For example, Google [9] crawls the Web and records how many links point to any given document. Simplifying the algorithms for the purpose of this exposition, Google considers a document with more links pointing to it as more valuable than one that is less linked to. The rationale is that if more authors of Web pages have felt it worthwhile to include a link, the document is in some way more valuable.

Similarly, one aspect of the SCAM system [10] searches the Web and finds collections of documents that are complete or partial mirrors of another site. We can view such mirror sites as a kind of structural feature of the World-Wide Web collection. Considering this collection-level structure, one might conclude that mirrored documents are more valued than documents for which such survivability precautions or performance enhancements are not undertaken.

As a final example, PHOAKS contains an element of collection structure analysis in that it will not count URLs as recommendations if the message that contains the URL has been posted to multiple news groups. The assumption is that such multiple postings are indicators for the message being an advertisement. Collection-level structure analysis can thus be used to determine the genre of individual documents.

2.3 Information Context

A third set of content-based filtering techniques attempts to determine the context in which documents are situated. One example of context is the *publisher* of the document, such as the New York Times, the National Inquirer, or the World-Wide Web Consortium. Another example for context is the *time* at which the document was published: a document published in the 1940s would for many purposes be a very different context than one published during the last two months. The context of a document can sometimes be determined straightforwardly by examining some readily available aspect of the document. For example, parsing the URL of a Web page might give a clear indication that the document is published by the New York Times. On the other hand, determining the main topic of a document to place it in a '*thematic* context', requires more indirect analysis. By a document's thematic context we mean the (semantic) topic it is covering. This is different from the document's genre, which refers to the document's surface form and style, independent of topic.

Once the context of a document has been determined and recorded in the collection metadata, the techniques discussed in this section assign a value based on the context. For example, documents from the New York Times might be valued higher than other documents that appear in an unknown publication context. Similarly, many information systems by default list newer publications first when displaying result sets. This is a kind of value-based filtering, with publication time as a temporal context to which value judgments can be attached. Of course, the decision of which context is better than another depends on the task at hand, or even on individual preferences. For a historian, older temporal contexts might be more valuable than recent ones, while a technology analyst might attach higher values to newer publications. Filter engines may therefore need to be customizable.

Contexts and their valuation can therefore be user-dependent. This is very much true in another example of context-based approaches. ReferralWeb [11] is a filtering system that requires searchers to register once. As part of this registration process, the system's metadata extractor searches for Web documents that contain the registrant's name, and then finds other names in the same documents that occur in close proximity to the registrant. Examples of names that would be found in this process are co-authors of papers, individuals who have participated in netnews exchanges, and links found on home pages. The system then recursively repeats the process once or twice for the 'related individuals' it found during the first step. Once this registration process is complete, the search filter is ready. ReferralWeb's collection metadata is a network graph. A user's 'context' consists of his 'community', the people directly or indirectly reachable in the graph, when starting at the node that represents the user.

ReferralWeb's filter engine prefers those documents that are somehow connected with anyone in the searcher's context. A document is assumed to be connected to the searcher if one of the individuals in his context is mentioned in the document, or is easily reachable from it[1].

Sometimes, systems place documents into contexts without automatically attaching any value to those contexts. Referring to Figure 1.1, metadata is extracted, but no filter engine

1.ReferralWeb uses this information for several purposes, such as finding experts to contact about questions. In the context of this discussion, the relevant use is the ranking of documents.

uses it. Instead, users can interactively control the metadata extractor, examine the extracted data themselves, and then act as human filter engines. The 'filter control' function in the User Interface oval of Figure 1.1 represents this activity. For example, Scatter/Gather [12], and SONIA [13] create contexts by attempting to automatically categorize documents into thematically related clusters. Users then examine the clusters and attach value to them by exploring some of the clusters more deeply, while discarding others.

Manually controlled clustering can be combined with automated filtering, if the clusters are used for relevance feedback. For example, SenseMaker [14] lets users control the metadata extractor interactively to vary the criteria used for iterative or recursive clustering. Users can alternatively cluster large result sets by criteria such as author, by similar contents, publication date, or common Web site. Clusters derived by choosing one of these criteria can then themselves be reclustered using another criterion. Once an interesting cluster is found through this interactive exploration, the cluster can be passed to the filter engine to retrieve more documents that would fit into the cluster.

Scatter/Gather's technique for creating thematic contexts is based entirely on statistics of word distributions, and it is intended to support browsing. COATER [15] instead explores how WordNet, an online lexical database, can be used to determine a document's *semantic* context in support of querying. The core of WordNet is a list of concepts, such as 'something written by hand'. For each concept, WordNet lists the words that are typically associated with that concept, such as 'handwriting', 'hand', and 'script'. Based on this information, COATER's metadata extractor constructs an index over its collection. The index includes not only the words that actually occur in the documents, but also other words that often occur in the same semantic context. When presented with a query, COATER's filter and search engines attempt to identify the documents whose main concepts overlap best with the main concepts of the query.

2.4 Document-Internal Content Tags

The final sub-category of content-based value filtering techniques involves explicit human initiative and maintenance. These approaches assign most of the analysis work not to a metadata extractor, but to the authors or publishers of the information. All of the approaches rely on human beings physically changing the documents inside the collections by adding mark-up tags that make statements about the document contents. Filtering engines then examine the markings and, depending on the filters' bias, draw value conclusions about the documents.

An important example in this class of approaches is the PICS system [16]. It was developed in response to concerns about sexually explicit material on the Internet. The idea is to have publishers add tags to documents. The tags would indicate what kind of contents the documents contained, with salient criteria being violence, nudity, profane language, etc. Given this preparation of the collection, filter engines can then prevent minors from viewing material deemed inappropriate by their parents.

PICS is actually only a special case of a more general set of metatagging approaches. Metatags are used as markers within documents to convey any information about the document's content. Some metatagging facilities are part of the HTML standard. A more extensive system for metatagging on the Web is being developed under the name of Resource Description Framework (RDF) [17, 18]. It allows the design of extensive data structures for metadata. The vision is to go beyond content rating, and to allow complex schemas to be built for Web sites. This is somewhat analogous to the schemas that describe the structure of databases. RDF is a framework for using metatags, not a definition of a particular set of tags. One could imagine value filtering facilities making use of these tagging systems. For example, the *ht://Dig* search engine which covers intranets takes advantage of hidden tags for ranking result documents.

All of these content-based techniques by definition rely on an analysis of static clues gleaned from documents or collections as a whole. The techniques surveyed in the following section instead rely on dynamic clues observed while the collections are being used.

3 Action-based Filtering Techniques

Action-based filtering techniques all work by observing human actions associated with a collection. Some techniques observe whether and how documents in a collection are accessed and manipulated. The access accounting module of Figure 1.1 represents this gathering of access statistics. Other techniques rely on human readers explicitly rendering judgments about the value of documents. In Figure 1.1 this kind of judgment is represented by the feedback function in the User Interface oval. The assumption behind all these techniques is that human beings are the most reliable judges of information value, and that the best approach to filtering is somehow to extract this judgment from them on a large

scale.

We distinguish action-based approaches by whether they receive explicit, intensional judgments from human users, or whether they gather value judgments implicitly, through conjecture based on observed user actions. Sections 3.1 and 3.2 describe the categories of techniques used for gathering judgment metadata explicitly and implicitly, respectively.

3.1 Explicit Judgment

The most direct means of gathering judgements is to have users enter explicit evaluations about documents, collections, or authors. This technique has been used for a long time. In the typical IR context of query/response interactions, explicit judgment is known as 'relevance feedback'. This concept was originally used for query refinement, rather than for the benefit of a filter engine module. Relevance feedback takes place when users retrieve documents that match a query, and then explicitly indicate which documents are relevant. On the basis of this relevant-set, the retrieval system then modifies the original query, and finds more documents that are hoped to be equally desirable.

3.1.1 Relevance Feedback for Filtering

As pointed out in [19], IR and information filtering are closely related, so it is natural that relevance feedback has been adapted for filtering tasks as well. For example, the Tapestry email repository and filter system [20] allows readers of email, netnews articles, or other information streams to annotate the documents they read. In terms of Figure 1.1, these annotations make up the judgment metadata repository. The filter engines of other readers can then extract documents based on these annotations.

Even though at the surface, relevance feedback and filtering are similar, Tapestry differs from traditional relevance feedback. For one, Tapestry's feedback mechanism uses judgments by more than a single reader. An arbitrary number of readers can comment on a document, and their collective judgment can be used in the filtering task. For example, users might ask to see only documents that received high marks from the colleagues in their department. Tapestry is thus an example of a *collaborative filtering* system. These are systems that rely on judgments from more than one source.

A second difference is that in Tapestry, the feedback itself is the grist for the filtering task: readers retrieve documents because the documents were annotated as highly valuable. In contrast, traditional relevance feedback systems operate by using the content of relevant documents as queries: similarity measures determine which documents in a collection are closely related to the documents in the relevant-set.

While Tapestry focuses on documents others recommend, the filter engines of Fab and GroupLens [21, 22], while also based on collaborative filtering, stress an additional element. They attempt to find out which users are best suited as sources of recommendations for a given user. They do this by comparing recommendations made by all users, and finding those users that have made similar recommendations or value choices in the past. For example, Fab consists of search agents that attempt to find useful information by roaming the Web. Each user maintains his own search agent. Based on the feedback a user provides in response to information gathered by his agent, the agent builds an interest profile. However, rather than relying solely on that direct feedback from its user, the agent communicates with other agents to find 'colleagues' whose users have similar interests. Recommendations by these like-minded agents are then used to find good documents.

3.1.2 Data-Triggered Filters

In data-triggered filters, the judgment metadata consists of manually constructed filter expressions. The most widely known examples of such systems are mail filter facilities which allow users to have filter engines automatically discard messages from particular sources, or with particular contents (for example [23]). In terms of our architecture (Figure 1.1), users enter filter expressions through the user interface into the judgment metadata repository. Filter expressions may control the filter engine of one user only, or they may be shared among the filter engines of multiple users (collaborative filtering).

Data-triggered filters are sometimes just like standing queries, and are thus related to focused search. But other uses make the relevance to value based approaches more clear. A more recent version of a judgment-based system, for example, that uses data-triggered filtering, is the commercial product NetNanny. It is intended to help users eliminate information they deem 'undesirable', based on their personal values. The filters can be applied to Web sites, news groups, chat rooms, and other sources of information. NetNanny enables users to construct arbitrary lists of words or phrases that are to trigger filtering activity.

Conversely, systems that provide selective dissemination of

information (SDI) can also be based on data-triggered technologies. One example is SIFT [24], which allows users to enter an interest profile into the judgment metadata. Filter engines match this profile against a stream of incoming netnews articles. Articles that match the profile are selected and forwarded to the appropriate user.

3.1.3 Synthesized Filters

Data-triggered filters rely on users to in effect write filter programs, or at least to provide explicit keywords. In contrast, synthesized filters introduce an abstraction between the filtering machinery and the user. Rather than specifying precise instructions to the filter, users of synthesized filters describe their task or current context. Based on these high level descriptions, the filter engine chooses among available filter technologies to configure itself. For example, the LyricTime system [25] contains a collection of about one thousand songs. LyricTime attempts to automatically select songs, and to play them on user workstations. The goal is for the filter engine to pick songs that are likely to please each individual user. The system does this by building judgment metadata in the form of user profiles. Like the judgment metadata of feedback systems in Section 3.1.1, profiles are constructed from explicit feedback to songs. The indirect nature of the system comes into play when the filter is used. LyricTime's user interface includes a 'mood' indicator. By choosing among 'cheerful', 'romantic', 'calm', 'sad', etc., users can declaratively control the filter engine's operation. LyricTime's judgment metadata actually contains multiple profiles for each user: one profile for each mood. This design helps the filter engine realize mood specific filtering.

3.2 Implicit Judgment

Filters that use explicit judgment mechanisms have the disadvantage of requiring user participation. Triggers must be installed, profiles need to be constructed, contexts are to be declared. These require time and attention from users. The approaches that have users explicitly express an opinion are especially notorious for problems with low participation.

The best way to overcome this obstacle is, of course, to reward users with such greatly enhanced filtering that they gladly invest the extra work. This is difficult especially for the case of collaborative techniques, where users sometimes fear that they might be supporting free-loaders. Even for strictly individual filter engines, the picture is complicated in that most explicit judgment filters improve only slowly over time. This delay in gratification can be a powerful disincentive to user participation.

In response to these problems, several systems have been developed that attempt to extract user judgments without requiring users to explicitly focus on an evaluation task. The goal is somehow to have the system observe user activities, and to conjecture users' opinions about documents from those observations. We distinguish between two classes of approaches to collecting implicit judgments: conjecture from observing collective user behavior, and conjecture from observing individual user behavior.

3.2.1 Judgment Conjecture from Collective User Behavior

One way to infer judgments about information is to observe how the majority of users interact with it. The most obvious approach is to analyze access logs and monitor how visitors browse a collection. In terms of Figure 1.1, the access accounting module gathers statistics on how many times particular documents in a collection are accessed. This information is stored in the judgment metadata repository. Usually, filter engines assume that frequently visited documents are more valuable than others.

The technique of analyzing access history logs for individual collections is effective only for collections with large enough traffic to generate statistically significant judgment data. Even with heavy traffic, the down side of approaches based on access logs is that they only provide judgment information about the respective collection. In the case of the Web, this problem arises because access logs are generally not shared among independently administered sites. Access logs can therefore inform filter engines only about relative judgments over documents within one Web site. Nevertheless, when a site contains many documents, filter engines can, for example, use this kind of access data for an important class of filtering: Guided tours.

Guided tours are programs that know about the resources at one site, and guide visitors through the site along a route that is hoped to be optimized for the visiting user's interest. While visitors might be expected to state their interest explicitly, the guiding process is controlled implicitly through observations of user behavior. Systems differ in how they obtain knowledge about visitors' interests, and in the methods they use to suggest links for visitors to browse. We examine some examples.

WebWatcher is one example for the use of site-specific access analysis in support of guided tours [26]. When enter-

ing a site that is covered by WebWatcher, a user enters keywords that correspond to his or her interests. As the user browses through the site, WebWatcher recommends links to follow whenever the user looks at a page. One of the methods the system uses to derive these recommendations is to record in the judgment metadata any observed correlations between links and interests: Whenever any user follows a link in the collection, that link, as well as the user's stated interest is recorded in the judgment metadata. When a Web page is presented to a new user, WebWatcher's filter engine ranks each link on the page. A link is ranked high if previous users who chose that link had similar interests as the current user.

Some systems avoid forcing users to enter interests explicitly. For example, in [27], an access accounting module accumulates as judgment metadata the sequence in which users have accessed the pages on a Web site. Based on these access paths, users are clustered into what is assumed to be groups of similar interests. The navigational path of a new visitor is monitored until the filter engine can make a guess as to which interest group the visitor might belong to. Thereafter, the filter engine makes suggestions on which paths the new visitor might wish to follow. For example, if most of the members in the new visitor's interest group examined a page on 'gloves' after arriving at a page on 'skiing', the filter engine would suggest the glove page when the new visitor enters the skiing page.

Another example in this category of judgment-implying systems is KSS [28]. This system works by having users perform all of their Web browsing through a proxy. The proxy records how many users followed any given link. When returning a Web page to the user, the KSS filter engine annotates each link within that page by adding the number of users that have followed that link in the past. The assumption, again, is that links which were followed more often are more valuable by some definition.

The Hotbot and Direct Hit search engines are experimenting with a similar technique. Like KSS before them, they collect their judgment metadata as follows. When presenting users with a list of search results, all the result URLs are made to point back to the search engine server. No matter which result the user elects to follow, the connection to the target is not established directly. Instead, a connection to the Hotbot/Direct Hit site is established first. The access accounting modules at those sites record which of the results the user selected. This judgment metadata can later be used to improve the filter engines' ranking operation.

None of the systems discussed in this section happen to suppress any information, although that might, of course, also be useful. They just annotate or rank information based on the available judgment metadata. In Figure 1.1 this use of judgment information is listed as one of the activities in the filter engine oval. We call this kind of filtering 'browse view generation'.

We note an important difference between the gathering of judgment metadata through collection-specific access log accounting, and the KSS or Hotbot approaches: KSS and Hotbot can collect judgment metadata for more than one site, because they are *portals* to many (Web) collections, rather than being an accounting module at one particular site.

Potentially, the portal approach is more powerful, because it can collect judgment metadata for a wide variety of information. The problem is that this advantage occurs only if enough people use the portal, just as local accounting for one collection is effective only if enough traffic visits that collection. The difference between portals and collections is that it is clear what collections need to do in order to attract traffic: They must provide important information or superior organization of the material they offer. Techniques for making a portal attractive are much less obvious: Portals must provide some value beyond the information they pass along. In the case of KSS, this additional value is the annotation of links. Another advantage might be that portals can more easily specialize their services to a particular population of users. For example, a portal installed for use by all the biochemists of a company might be able to provide particularly good service for biochemists, because its judgment metadata will be dominated by this relatively homogeneous population.

The issue is further complicated by commercial considerations. Information providers often derive financial gain from users visiting their site directly, rather than through a portal. The detailed reasons involve advertising revenues, and are not of central concern here. The effect, however, is that portals sometimes run into difficulties because information providers object to the traffic indirection portals represent.

3.2.2 Judgment Conjecture from User-Specific Behavior

While judgment metadata supporting techniques in the previous section is based on observations of collective user behavior, techniques discussed now attempt to collect judg-

ment metadata about a particular user's interests. Filter engines that operate with user-specific judgment metadata can evaluate any document, even if neither the engine's user, nor potentially any other users have ever seen that document. In contrast, filter engines in the previous section were able to function only if documents they evaluated had previously been visited by other users.

The main obstacle to overcome for this set of approaches is the collection of judgment metadata for each individual user. Two primary sources for this metadata are observations of which documents a user accesses, and observations of what the user does with the documents he views.

As an example for access based observations, recall the earlier facility where an access accounting module in a portal observed how many times users selected a given URL from a list of search results. This technique can be used for user-specific judgment conjectures as well. This time, the target document is associated with a particular user, not just with an access count. This method can be refined in that the judgment metadata could go beyond recording the fact that the user accessed the document. The words in the document could also be recorded as potential keywords of interest. Another variation is to observe how often a user visits not an individual document, but a particular collection. For example, if the user frequently consults the online Encyclopaedia Britannica, then any document within that collection could be taken to be valuable for that user.

The second set of judgment metadata ideas involves identifying observable user behaviors that are effective predictors of user interest. One predictor is the time a user spends studying a particular document. One study [29] shows that time spent reading a particular netnews article is indeed a good indicator of interest. It further shows that time not spent is a good indicator of disinterest. Finally, the study suggests that interest or disinterest in a piece of information is a good predictor for levels of interest in closely related information, such as follow-on messages in a newsgroup discusssion thread. This study was controlled to avoid users being distracted by other activities. In practice, time actually spent on a document can be difficult to monitor.

Other related ideas for user-specific judgment metadata are based on the assumption that, except for deletion, users will manipulate documents only if they are interested in them. Manipulations might include making a bookmark to the document, saving it on a local disk, linking to it or, in the case of message streams like netnews, replying to a message (c.f. Tapestry). Similarly, the act of following links from a given page might indicate interest in the page that contains those links.

Such user action metadata collection can be accomplished through control over the user interface. By inserting an 'observer' in the interface, the user's actions can be monitored and recorded. In Figure 1.1, this fact is represented by the arrow from the user interface to the judgment metadata repository. For example, systems that include control over a user's browser, or have access to the user's bookmark files can observe these behaviors.

Collecting user-specific access metadata can, of course, also be accomplished by modifying a user's interface to the documents. Some judgment metadata gathering approaches, however, attempt to avoid this technique, because it requires software distribution, support, and maintenance. In considering alternative solutions, notice that collection-specific access accounting modules of the previous section enjoyed the 'luxury' of being situated at one central place with the collection they monitored. Judgment metadata collectors discussed here need to follow one user anywhere he goes.

One way to accomplish such external access accounting is to use the portal technique. However, while the use of portals for collective user techniques can operate even if users remain anonymous, the single-user case requires that users be persuaded to reveal all their Internet destinations to the organization that operates the portal. This organization can then promise to provide an effective filter engine that is customized to the user's need. One related experiment under way commercially is to offer free Internet access in return for the right to interject an access accounting module between the user's computer and the Internet. The commercial interest behind this experiment is, of course, targeted advertising.

Once user-specific judgment metadata is available, filter engines can use it in several ways. For example, Siteseer [30] compares the bookmarks of multiple users, and determines how much overlap exists among the bookmark sets. Users that tend to establish bookmarks to the same pages are assumed to have common interests. Based on these user profiles, Siteseer will prefer pages that have been bookmarked by users that have interests in common with the searcher. Note that this example gathers user-specific behavior, and aggregates this information to support collective filtering. If user-specific judgment metadata is available for multiple users, such 'collectivization' is always an option.

Like WebWatcher, Letizia's filter engine [31] uses its understanding of individual users to evaluate the merits of links on a Web page. The evaluation is based on standard IR techniques that match those pages with the user's interest, as represented by a vector of keywords. In choosing among links on a page, both systems recommend the ones that lead to the most rewarding destinations down the line.

User-specific judgment metadata can, of course, also be used to construct interest profiles for SDI activities of the filter engine in Figure 1.1. This can be particularly effective if judgment metadata is continuously updated and corrected.

4 Discussion and Summary

Figure 4.1 is a graphical summary of the value filtering techniques we have discussed. There are many purposes for

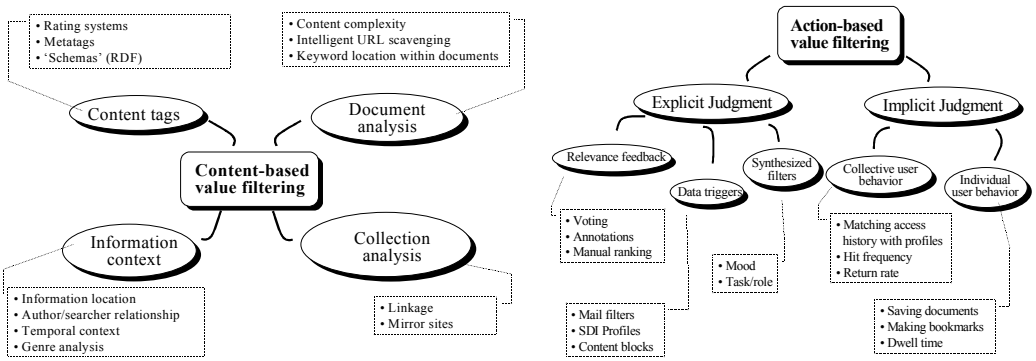

Figure 4.1: Summaries of content-based and action-based value filtering

which value filters can be used. We have touched mostly on applications that directly benefit searchers. The most frequent use in this context is the blockage of irrelevant information, and the ranking of documents to help users digest large result sets. In addition, value filters can improve browsing activities through guided tours. Services that selectively disseminate information, and facilities that notify users of document changes can also be improved with value filtering technology.

Users of information are, however, not the only beneficiaries of value analysis. Large-scale information providers can benefit as well. The most prevalent use so far seems to have been the improvement of advertising accuracy. But the technical performance of information servers can also be improved with the help of collection and judgment metadata. For example, caching, or the organization of indexes can be guided by the knowledge contained in the metadata. Similarly, the acquisition of new information, such as the operation of crawlers, can be informed by an understanding of information value. For example, when limited in time and processing resources, crawlers can revisit high-value documents more often, or can explore high-value sites more deeply than other documents and sites that appear to be less important [10].

A wide variety of work remains to be accomplished in the area of value filtering. There is, of course, room for invention of new types of collection- and judgment metadata. Similarly, novel techniques of designing the corresponding filter engines would help. There are also some broader, very promising additions that need to be explored. For example, notice that in Figure 1.1 there is no arrow between the filter engine and the search engine. One could argue that this is an important shortcoming in current systems. In current systems, users can search over documents, which are then processed by the filter engine. The filter engine potentially modifies the documents, maybe by adding link visit frequencies. Users should be able to include this information in their queries. For example, a user should be able to search for documents that contain certain keywords, *and* contain links that are visited frequently. This connection between filter engines, search engines, and query facilities is quite unexplored.

More generally, recall that Figure 1.1 is a conceptual architecture, not the plan of any single current system. Quite a bit of progress could presumably be made if all the elements of this architecture were combined into a one operational facility.

The multiplicity of approaches to value filtering invites several analyses we cannot undertake in this survey. One important analysis would examine the dynamic characteristics of

the filtering approaches we discussed. For example, filters can suffer from the effects of positive feedback loops. Consider, for instance, a filter engine that always presents the highest valued information first. In the case of a search result list, this information would be a link. If clicking on a result is interpreted as the user's implicit positive value judgment, then a positive feedback loop can occur in this scenario. The reason is that many users have a natural tendency to select the first entry in any list. Once a document is listed at the top, this phenomenon would tend over time to skew the judgment metadata, because the physical position, not a real value judgment, would lead to continued strengthening of the document.

Another important analysis to undertake is the impact of the various value filtering techniques on privacy. Clearly, some of the techniques we described are more intrusive than others. Any metadata collection that correlates individual users with the documents they obtain has obvious implications for privacy. But even among these techniques, subtle differences and ethical gradations need to be examined. One example is disclosure. For example, most technology savvy users of search engines have long understood that their queries could easily be recorded and associated with their internet address. The recent experiments with collecting judgment metadata by monitoring which result the user selects after a search, takes this exposure one step further. One of the questions that arise is whether this collection activity needs to be prominently brought to the user's attention.

Yet another analysis of central importance is one of comparative filter effectiveness. Many of the techniques discussed here have not been thoroughly analyzed and compared. For example, it seems unclear how heavily the accuracy of using users' browsing routes through a site as implicit judgment metadata is affected by the location of links within the browsed documents. For example, maybe the links near the top of documents are more likely to be chosen just because we tend to scan documents from top to bottom. We might get enticed by an early link, and then are so distracted by what we find that we never return to the starting document, even though links listed later in the document might have been much more valuable.

It also seems unclear how effectively sampling techniques might perform for the filtering task. For example, in the case of techniques for implicit judgment collection, is it sufficient only to use a subset of users to determine value accurately? If so, should the subset of users be of a particular distribution, including contributors from educational institutions, commercial entities, or different countries of origin? Similarly, is it sufficient to determine the value of an entire collection, and to use that value for all the documents contained in that collection? Or does the value of each document within the collection need to be evaluated separately to ensure satisfactory filtering accuracy? How does the answer to this question depend on the genre of the documents in the collection? How often do value judgments need to be recalculated? How quickly does value decay, either at the level of individual documents, or at the granularity of entire collections?

Finding the answers to these questions will require experiments and thought. As the Web and other collections grow, and as new media and document genres are penetrating into the online world, traditional text-based search techniques need to be augmented. For this reason, value filtering is increasingly gaining in importance. While this discussion has pointed out several promising classes of solutions, many questions remain to be answered. Exciting experiments are in progress, both to help find the answers, and to break new ground. More such experiments, and careful evaluations are waiting to be undertaken.

5 References

[1] Stefano Mizzaro. Relevance: The Whole History. *Journal of the American Society for Information Science*, 48(9):810–832, 1997.

[2] Linda Schamber. Relevance and Information Behavior. *Annual Review of Information Science and Technology (ARIST)*, 29:3–48, 1994.

[3] Shoshana Loeb and Douglas Terry. *Information Filtering*. Communications of the ACM, December, 1992.

[4] Paul Resnick and Hal R. Varian. *Recommender Systems*. Communications of the ACM, March, 1997.

[5] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter. PHOAKS: a system for sharing recommendations. *Communications of the ACM*, 40(3):59-62, March, 1997.

[6] Marti A. Hearst. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In *Proceedings of the Conference on Human Factors in Computing Systems*, 1995.

[7] Brett Kessler, Geoffrey Nunberg, and Hinrich Schuetze. Automatic Detection of Text Genre. In *Proceedings ACL/EACL*, 1997.

[8] Sergey Brin. Extracting Patterns and Relations from the World Wide Web. In *WebDB Workshop at 6th International Conference on Extending Database Technology,*

EDBT'98, 1998. Available at http://www-db.stanford.edu/ sergey/extract.ps.

[9] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the Seventh World-Wide Web Conference*, 1998.

[10] Junghoo Cho, Narayanan Shivakumar, and Hector Garcia-Molina. Computing Document Clusters on the Web. In *Submitted to VLDB '99*, 1998.

[11] H. Kautz, B. Selman, and and M. Shah. Referral Web: Combining Social Networks and collaborative Filtering. *Communications of the ACM*, 40(3):63-65, March, 1997.

[12] Douglass R. Cutting, Jan O. Pedersen, David Karger, and John W. Tukey. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 318-329, 1992.

[13] Mehran Sahami, Salim Yusufali, and Michelle Baldonado. SONIA: A Service for Organizing Networked Information Autonomously. In *Proceedings of the Third ACM International Conference on Digital Libraries*, 1998.

[14] Michelle Q Wang Baldonado and Terry Winograd. SenseMaker: An Information-Exploration Interface Supporting the Contextual Evolution of a User's Interests. In *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 11–18. ACM Press, New York, Atlanta, Ga. March, 1997.

[15] Mark A. Stairmand. Textual Content Analysis for Information Retrieval. In *Proceedings of the Twentieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1997.

[16] P. Resnick and J. Miller. PICS: Internet Access Controls Without Censorship. *Communications of the ACM*, 39(10):87-93, 1996.

[17] Eric Miller. An Introduction to the Resource Description Framework. *D-Lib Magazine*, May, 1998. http://www.dlib.org/dlib/may98/miller/05miller.html.

[18] *Resource Description Framework (RDF) Model and Syntax Specification*. Number WD-rdf-syntax-19980819. World-Wide Web Consortium, 1998. Available at http://www.w3.org/TR/WD-rdf-syntax/.

[19] N.J. Belkin and W. Bruce Croft. Information filtering and information retrieval: two sides of same coin? *Communications of the ACM*, 35(12):29-38, December, 1992.

[20] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35(12):61-70, December, 1992.

[21] M. Balabanovic and Y. Shoham. Fab: content-based collaborative recommendation. *Communications of the ACM*, 40(3):66-72, March, 1997.

[22] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, and J. Riedl. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40(3):77-87, March, 1997.

[23] Thomas W. Malone, Kenneth R. Grant, Kum-Yew Lai, Ramana Rao, and David Rosenblitt. Semistructured Messages are Surprisingly Useful for Computer-Supported Coordination. *ACM Transactions on Information Systems*, 5(2):115-131, April, 1987.

[24] T. Yan and H. Garcia-Molina. SIFT—A tool for Wide-Area Information Dissemination. In *Proc. 1995 USENIX Technical Conference*, pp. 177-186, New Orleans, 1995. http://sift.stanford.edu.

[25] S. Loeb. Architecting personalized delivery of multimedia information. *Communications of the ACM*, 35(12):39-48, December, 1992.

[26] T. Joachims, D. Freitag, and T. Mitchell. WebWatcher: A Tour Guide for the World Wide Web. In *Proceedings of IJCAI97*, 1997.

[27] Tak Woon Yan, Matthew Jacobsen, Héctor García-Molina, and Umeshwar Dayal. From User Access Patterns to Dynamic Hypertext Linking. In *Proceedings of the Fifth World-Wide Web Conference*, 1996.

[28] Andreas Paepcke, Hector Garcia-Molina, and Gerard Rodriquez. Collaborative Value Filtering on the Web. KSS. In *Proceedings of the Seventh World-Wide Web Conference*, 1998.

[29] M. Morita and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[30] J. Rucker and M.J. Polanco. Siteseer: personalized navigation for the Web. *Communications of the ACM*, 40(3):73-75, March, 1997.

[31] Henry Lieberman. Letizia: An Agent that Assists Web Browsing. In C.S. Mellish, editor, *Proceedings of 14th International Joint Conference on Artificial Intelligence*, 1995.