A Methodology for Intranet Search Engine Evaluation

Dick Stenmark
it.dixi@memo.volvo.se

Volvo Information Technology
Dept. 9530, HD1N
SE-40508 Göteborg, Sweden

IT & Organisation
Viktoria Institute
Göteborg University
Box 620
SE-40530 Göteborg, Sweden

Abstract

Although several comparative studies of internet search engines have been published, few have focused on the specific considerations that we have found an intranet tool to require. While other evaluations have provided soon outdated answers, this paper instead discusses what questions to ask when analysing and evaluating intranet search tools. The discussion takes departure in the evaluation process of selecting a search tool for a large corporate intranet, and is refined through repeated interviews and discussions with people covering a variety of organisational roles. We conclude that most other evaluations only consider a small, predefined set of features that are likely to be insufficient for a given installation, and we therefore suggest a somewhat new approach that adds a dimension to the evaluation process. The proposed methodology has the advantages of being product independent, being able to age more gracefully, and being able to spot strength and weaknesses more easily.

Keywords: Intranet, search engine, evaluation, methodology

BRT Keywords: AB, CD, EF

Introduction

As the World Wide Web (WWW) started to explode in terms of users, servers, and pages, it became obvious that search capabilities had to be added, and a flourishing market of public search engines emerged. Later, the growing number of *intranets*, i.e. intra-organisational webs hidden from the internet behind firewalls and proxies, has created a need and a market not only for web search *services* but also for web search *products*. Many commercial search engines and spiders are available on the market, but how do we know which best suit the needs and requirements of our organisation?

Though many comparative search engine studies have been published, (e.g. Chu

and Rosenthal, 1996; Goldschmidt *et al*., 1997; Lange, 1997; Leighton and Srivastava, 1997; Nance, 1997; Notess, 1995; Slot, 1995), the majority have evaluated public *internet* search engines. Comparing an *intranet* search tool evaluation (e.g. Goldschmidt *et al*., 1997) with a public search tool evaluation (e.g. Chu and Rosenthal, 1996), you can see that, though some features are common to both, the former has many criteria *not* considered by the latter (e.g. operational aspects). This observation raises the question whether or not intranet search engines should be evaluated differently than public search services. We argue it should.

One thing the above seven evaluations have in common is that they all present *answers*, though these are bound to be outdated fairly soon after publication. This is particularly true for precision studies (Leighton and Srivastava, 1997), but also for most other observations. Previously published studies all compare pre-selected products relative to each other and the answers found depend on the tools being studied. A novelty with this paper is that it discusses issues relevant for intranet search engines in particular, without investigating specific products. The facts that new tools or new features keep appearing, and that the performance of these tools and features are changing, indicates that the answers themselves are not useful to collect beforehand. Instead, the focus should be on what *questions* to ask, since the questions are likely to be relevant longer than the answers.

Having studied two organisations implementing intranet search capabilities, we advocate an evaluation methodology that takes departure in the actual needs of the organisations doing the evaluation. Different organisations have different needs, and since we recognise that these needs cannot be defined beforehand, but as part of the evaluation process, our methodology is applicable to any organisation. Other evaluation methods have been used elsewhere to rank particular products, without being explicitly suggested or announced as general methodologies (e.g. Goldschmidt *et al*., 1997, Nance, 1997). Chu and Rosenthal (1996) do however present a general methodology that could be useful, but their aim is again to identify the best *public* search engine.

There are four advantages to our methodology:

1. The focus is on intranet search engines and their particular requirements rather than on public search services, as with most other comparative studies
2. The organisation doing the evaluation selects exactly those features that are important to their situation and needs instead of having to rely on a predefined set of evaluation criteria.
3. Since the features are not equally important, they are grouped and weighted according to importance to better balance the final result.
4. By summarising not only the individual features, but also per area of functionality, a greater understanding of the products strong and weak sides is gained.

There are two types of tools to consider; general Information Retrieval (IR) tools that address the whole spectra of electronic information that might be found in a company, including video, images, and databases, and more specific web tools that focus on HTTP-accessible documents only. Of course, the former is much more complex and requires much more consideration than does the latter, even if only the web indexing part is considered. Our research has focused on web indexing only.

The next section discusses the settings and the research method used, and we thereafter defined five areas of functionality. We then look into these areas more closely, discussing the different features in section 3. In section 4 we develop the evaluation procedure, and section 5 contains the summarising conclusions.

Method and Domain

Our approach to research has a phenomenological orientation and we have thus used mainly qualitative methods while collecting the data. Instead of collecting hard data from log files, we have interviewed *people*. During a longitudinal study of company A, including six months of fieldwork, we gained an understanding of what people in different positions in the company thought about web searching. This ethnography-informed approach had the advantage of being good at providing both explanations and new insights. Since our focus has been on the *use* of technology, we have tried to develop a method useful to practitioners rather than produce some theoretical framework.

During our research, which took place in 1997 and 1998, we participated in the processes of implementing intranet search capabilities at two large companies. However, the major part of the work was conducted at company A, which had approximately 25,000 users connected to an intranet consisting of some 200 servers. We were able to work closely with the project members and participated in the project all the way from the very beginning to the final decision. During these six months we interviewed 14 people, including managers at different levels, IR personnel and librarians, people from operations, content providers, computer scientists, and end-users. Many of these people were interviewed several times, and we spent a total of 42 hours on individual interviews. We also arranged three larger meetings and two workshops to which interested parties were invited to debate different needs and try various design suggestions. In parallel, we did a thorough literature study to see how others had done their evaluations. Ideas and methods found in the literature were discussed during the interviews and meetings, and strong and weak points noted. At the end of the project, we analysed the collected material, noticing what approaches had been successful. This experience was generalised to form a tentative methodology.

Our contacts with company B, which has some 80,000 intranet users, were limited to the project manager, with whom we conducted several telephone interviews and exchanged email. Company B had a similar intranet search project running and we sent the project manager our methodology draft for him to test and comment. We were also given access to their evaluation protocols. Through this process, we received insights that helped further refine the methodology.

A later draft of the methodology was then evaluated by Corporate Express.

Areas of functionality

Based on interviews and literature studies, we identified 80 different criteria. This large number of features, and the spread of areas to which they belong, made it difficult to get an overview of the evaluation. We therefor divided the features into separate areas of functionality. Though there are several ways to do this, the perhaps most obvious one is to define them as *a spider*, which crawls through the network, *an index*, where the data is managed, and *a query handler*, which serves as a front-end for the users. However, these three categories alone are not sufficient, since they do not contain all the identified aspects.

The remaining features might then be collected in a miscellaneous or *general* section, as suggested by Goldschmidt *et al.* (1997), but other options are also available. For example, Chu and Rosenthal (1996) used five areas; index, search capabilities,

performance, output, and user effort. Their view of index is the same as ours, while we collect their search capabilities, output, and user effort in our query handler area. Performance cannot be measured prior to installation, and since Chu and Rosenthal are concerned with public engines only, they do not consider operational aspects. Having taken this into account we suggest the following five areas:

1. *Platform, scalability, and support* should cover the type of hardware and software the product runs on, how the pricing is set, and what support is available.
2. *Data gathering* deals with the spider and its particular features.
3. *Index* is about how the index is organised internally.
4. *Search features* include search capabilities, interface issues, and user documentation.
5. *Operation and maintenance* cover both administrative aspects and pure operational issues.

Having defined the above five areas, we associated each aspect suggested by the interviewees with the corresponding functional area. The identified features will now be briefly discussed.

Platform, scalability, and support

This area is the one most specific to intranets and thus, despite its importance, never considered in internet search engine evaluations. Start by checking if the product is available on many platforms or if it requires the organisation to invest in new and unfamiliar machinery. An *intranet* search engine often runs on a single machine and is operated and maintained by people with knowledge about servers, but not necessarily experts in spider technology. This suggests that a good intranet spider should be designed specifically for an intranet and not just be a ported version of an internet spider. Still, the spider and the index must be able to handle a fairly large amount of data without letting response times degrade or the users will be upset. For example, a product that can take advantage of multi-processor hardware scales better as the intranet grows. The product should therefore have been tested to handle an intranet of the intended size.

It is difficult to estimate data collection time since it depends on the network, but during the test installation this activity should be clocked. If the intranet consists of one server only, a spider is not needed, but as the web grows crawling capabilities become essential. A spider allows for the net to grow without forcing the webmasters to install indexing software locally. Also, try to determine how query response times grow with the size of the index. If an index in every city, state, or country where the organisation is represented is wanted, make sure the product supports this kind of distributed operation.

Having technical support locally is an advantage if the local support also has local competence. If questions have to be sent to a lab elsewhere, the advantage is lost. Examine what kind of support is offered and to what cost. Sometimes installation and training are included in the price. How long the products have been available and how often they are updated are important factors that indicate the stability of the product. It might also be important to ask about future plans and directions. The above discussion is summarised in table 1 below.

Table 1: Questions about platform availability, scalability of components, and support.

1. What platforms does the product run on (HW and SW)?
2. Is the product designed for intranet?
3. Can the product exploit multi-processor HW?
4. How large data volumes can be handled (in Mb)?
5. How long will data collection take (per Mb)?
6. Can remote web servers be indexed?
7. What is the upper limit for query response times?
8. Can the load be distributed over several servers/indices?
9. Is some sort of bandwidth-saving technique utilised?
10. Where is the nearest technical support?
11. Does technical support cost extra?
12. What is included in the price? Installation? Training? Upgrades? Support?
13. Is the price scaleable?
14. How long has the product been available?
15. How often is the product updated?
16. What is coming within the next 6-12 months?

Data gathering

Most organisations have legacy data in formats other than HTML, e.g. Adobe's PDF, MS-Office, FrameMaker, Lotus Notes, Postscript, and plain ASCII text. The spider should at least be able to correctly interpret and index the most frequently used or the most important of these formats. If meta tags and XML tags are likely to show up within the documents, the spider must be able to interpret such tags. If USENET newsgroups need to be indexed, the spider must be able to crawl through them. That also goes for client side image maps, CGI-scripts, Lotus Domino servers, and frames. Frames are supported by most web browsers and frequently used within many companies. Spiders generally work their way round the net by picking up and following hypertext links, but a page with frames has another syntax for links that might cause the spider to misinterpret them, or simply ignore them.

Intelligent robots are able to detect copies or replicas of already indexed data while crawling and advanced search engines can index "active" sites, e.g. sites that update frequently, more often than more "passive" sites. If this is not supported, some manual means of determining time-to-live should be provided. There should be some means of restricting the robot from entering certain areas of the net, including any desired domain, subnet, server, directory, or file level. Also, check if search depth can be set to avoid loops when indexing dynamically generated pages. Support for proxy servers and password handling can be useful, as can the ability to not only follow links but also detect directories and thus find files not linked to. The spider should be easy to set up and start. Check how the URLs from which to start are specified as well as if the users may add URLs.

The Robot Exclusion Protocol (Koster, 1994) provides a way for the webmaster to tell the robot *not* to index a certain part of a server. This should be used to avoid indexing temporary files, caches, test or backup copies, as well as classified information such as password files. The above discussion is condensed in table 2 below.

Table 2: Questions about the data gathering process.

1. What formats other than HTML can be indexed by default?
2. Are any APIs (or equivalents) available to add more formats?
3. Can meta data be indexed?
4. Can ad-hoc XML tags be handled?
5. Can USENET News be indexed?
6. Can image maps be handled?
7. Are CGI scripts handled?
8. Can Notes Domino servers be handled correctly?
9. Can frames be handled correctly?
10. Are duplicate links automatically detected?
11. Can the spider determine how often to revisit a site/page?
12. Can Time-to-live be specified?
13. How can the crawling be restricted?
14. Can the search depth be specified?
15. Can the product handle proxy servers?
16. Does the product handle password protected servers?
17. Can the spider find directories on its own?
18. Is the spider easy to set up and control?
19. Can users add URLs?
20. Does the spider honour the Robot Exclusion Protocol?

Index

Though a good index alone does not make a good search engine, the index is an essential part of a search tool. Chu and Rosenthal (1996) conclude that one of the most important issues is keeping the index up-to-date, and the best way to do that is to allow real-time updates. There is a big difference between indexing the full text or just a portion. Though partial indexing saves disk space it may prevent people from finding what they are looking for. The portion of text being indexed also affects the data that is presented as the search result. Some tools only show the first few lines while others may generate an automatic abstract or use meta-information.

If the organisation consists of several sub-domains, users might only want to search *their* sub-domain. Allowing the index to be divided into multiple collections might then speed up the search. Some tools support automatic truncation or stemming of the search terms, where the latter is a more sophisticated form that usually performs better. If the organisation is located in non-English speaking countries the ability to correctly handle national characters becomes important. Also, note that some products cannot handle numbers. If number searching is required, e.g. serial numbers, take this limitation into consideration. Should words that occur too frequently be removed from the index? Some engines have automatically generated stoplists, while others require the administrator to remove such words manually.

Search engines are of little use if an overview of the indexed data is wanted, *unless* they are able to categorise the data and present that data as a table of content. Automatic categorisation may also be used to focus in on the right sub-topic after having received too many documents. If information about when a particular URL is due for indexing is available, it is useful to make it accessible to the user. Table 3 contains a summary of the above discussion.

Table 3: Questions about the index and how the index is structured internally.

1. Is the index updated in real-time?
2. Is the full text or just a subset indexed?
3. Can abstracts be created automatically?
4. Can the product handle several collections?
5. Does the product exploit stemming?
6. Can the product correctly index national characters?
7. Are numbers indexed correctly?
8. Are stoplists for common words supported?
9. Is there any built-in support for automatic categorisation?
10. Does the index have an URL status indicator?

Search features

The user query and the search result interfaces are often sadly confusing and unpredictable. Shneiderman *et al*. (1998) argue that the text-search community would greatly benefit from a more consistent terminology. Since we do not yet have this commonality, evaluation of the search features must be done with great care. Different vendors use different names for the same feature, or the same name for different features.

Though Boolean-type search language is often offered, most users do not feel comfortable with Boolean expressions (Hearst, 1997). Instead, studies have shown that the average user only enters 1.5 keyword (Pinkerton, 1994). Due to the vocabulary problem (Furnas *et al*., 1987), the user is likely to receive many irrelevant documents as a result of a one-keyword search. Natural language queries have been shown to yield more search terms and better search results, even when performed by skilled IR personnel (Turtle, 1994). Apart from Boolean operators, a number of more or less sophisticated options (e.g. full text search, fuzzy search, require/exclude, case sensitivity, field search, stemming, phrase recognition, thesaurus, or query-by-example) are usually offered. One feature to look for in particular is proximity search, which lets the user search for words that appear relatively close together in a document. Proximity search capability has been noted to have a positive influence on precision (Chu and Rosenthal, 1996).

Many organisations prefer to have a common "company look" on all their intranet pages. This requires customisation that may include anything from changing a logo to replacing entire pages or chunks of code. Again, this is an aspect irrelevant to public search services but something an intranet search engine might benefit from. Sometimes a built-in option allows the user to choose a *simple* or an *advanced* interface. It should also be possible to customise the result page. The user could be given the opportunity to select the level of output, e.g., by specifying *compact* or *summary*. Further, search terms may be highlighted in the retrieved text, the individual word count can be shown, or the last modification date of the documents may be displayed. It can also be possible to restrict the search to a specific domain or server, or to search previously retrieved documents only. For the latter, relevance feedback is a very important way to improve results and increase user satisfaction (Shneiderman *et al*., 1998).

Ranking is usually done according to relevancy of some form. However, the true meaning of the ranking is normally hidden to the user, and only presented as a number or percentage on the result page. More sophisticated ways to communicate this important information to the user have been developed (e.g. Rao *et al*., 1995), but not many of the commercially available products have yet incorporated such features. However, the possibility to switch between relevancy and date is often supported. Dividing the results

into specific categories might help the user to interpret the returned result. Finally, does the product come with good and extensive online user documentation? The above discussion results in the questions in table 4.

Table 4: Questions about the search interface and the features available to the users.

1. Can Boolean type queries be asked?
2. Are true natural language queries supported?
3. Does the product support full text search?
4. Is fuzzy matching supported?
5. Does the product support + (require, MUST) and - (exclude, MUST NOT)?
6. Can case sensitivity be turned on/off?
7. Can attributes or fields be searched?
8. Is stemming done automatically?
9. Does the product recognise phrases?
10. Does the index utilise a thesaurus?
11. Is "query by example" supported?
12. Does the product have proximity search (NEAR)?
13. Can the search interface be customised?
14. Can the layout of the result page be customised?
15. Are there both Simple and Advanced interfaces?
16. Can the user select the level of output (compact, normal, summary)?
17. Are search words highlighted in the results?
18. Does the product show individual word count?
19. Does the index display modification date per document?
20. How can the search be restricted? (By date? By domain? By server? Not at all?)
21. Can previous results be searched (refined search)?
22. Is relevance feedback implemented?
23. Is the ranking algorithm modifiable?
24. Can the result be ranked on both relevancy and date?
25. Does the product have good on-line help and user documentation?

Operation and maintenance

As with area 1; *Platform, scalability, and support*, the operation and maintenance area is of no importance to public search engine evaluations and thus left uncommented. For an internal search service this area is of course highly interesting. We found great differences in how straightforward the products were to install, set-up, and operate. Some required an external HTTP server while others had a built-in web server. The latter were consistently less complicated to install. However, installation is probably something done once while indexing and searching is done daily. This ratio suggests that indexing and searching features should be weighted higher than installation routines. Running the spider should not interfere with how the index is operated. Both these components need to be active simultaneously.

An important feature is the ability to automatically detect links to pages that have been moved or removed. If dead links cannot be detected automatically, the links should at least be easy to remove, preferably by the end-user. Allowing end-users to add links is a feature that will off-load the administrator. Functions like email notification to an

operator should any of the main processes die, and good logging and monitoring capabilities, are features to look for. We found that products with a graphical administrator interface were more easily and intuitively handled, though the possibility of being able to operate the engine via line commands may sometimes be desired. It should also be able to administer the product remotely via any standard browser. Documentation should be comprehensive and adequate. The above is summarised in table 5.

Table 5: Questions about operational issues and maintenance of the product.

1. How easy is the product to install and maintain?
2. Does the product come with an embedded HTTP server?
3. Can the spider and the index be run independently?
4. Can old/bad links be removed easily or even automatically?
5. Does the product have email notification?
6. What logging/monitoring features are included?
7. Is there a graphical Admin interface?
8. Can the product also be operated in line mode (e.g., from a script or cron-tab entry)?
9. Can the product be administered remotely?
10. Is there enough documentation of good quality?

The evaluation method

Checking a large number of features for a number of candidates would be unnecessarily time consuming. During our interviews we noticed that the majority of the above features were categorised as either unimportant or just nice-to-have. The main point advocated here, and what seems to be a novel observation, is that *what features are important and what features are not, cannot be decided in general, but must be determined for each site or installation separately.*

The use of precision and recall goes back to 1955 when Kent *et al.* proposed them as the primary (and sole) measures of performance (see Saracevic, 1995). Ever since, those two criteria have been used in many search engine evaluations. However, both precision and recall require the tools to be installed before these quantities can be measured. This is not useful when the very objective is to determine *what* tools to install. Raghavan *et al*. (1989) question the usefulness of precision and recall as measures of IR systems performance and argue that in real life items are more or less relevant, and that precision requires a definition of what is to be considered as relevant. Saracevic (1995) notices that relevance is a complex human cognitive and social phenomenon, and argues against binary yes-no decisions. Nielsen (1999) points out that precision and recall assumes that the users want a complete set of relevant documents. This might have been true in traditional IR, Nielsen argues, but on the Web nobody will have time to read all relevant documents, so it is more important to present a small but useful sample.

For these reasons, we believe that precision and recall should be left out at this stage of the evaluation. Chu and Rosenthal (1996) point out that recall as an evaluation criteria for web searching is problematic, since it is almost impossible to know exactly how many relevant documents there are for a specific query. It may however still be useful to do precision tests *after* more than one product have been selected and installed, since one can compare relative recall between systems.

Weighting what matters

To reduce the above attributes to a more manageable number, the organisation must decide which features are important. First, are there any absolutely essential requirements? If products cannot even be considered without having a particular feature, this should be established and agreed upon early, and candidates not fulfilling the requirements should be eliminated.

A reference group should go through the list and select only those features that are important to their operation. At the two sites studied, we found that 32 and 38 selected features, respectively, were sufficient. These numbers are considerably larger than the 15 suggested by Chu and Rosenthal (1996) and the (other) 15 used by Goldschmidt *et al.* (1997), and should therefore give a more balanced picture. *The selected features are, however, not equally important.* We noted that people stressed certain features as being "crucial", "necessary", or "very important" while referring to other features as "good", "nice", useful", or "feasible". This led us to divide the features into categories according to importance.

Though both organisations in our study chose three categories, the number is individual and arbitrary, and should be set according to needs. We assigned weights to each category with the rationale being that important features should have greater impact on the overall result. The process of assigning weights involved the whole reference group, since no one person alone had competence enough, and since a group decision is likely to be less biased and should thus yield a more correct result.

For weighting to be successful there should be some significant weight difference between attributes that are significantly more important and attributes that are not. For example, say there are three categories, *Very important*, *Important*, and *Useful*, and that each item in the *Very important* category gets the weight 8, each item in the *Important* category gets 4, and *Useful* items get 2. Such a weighting means that a feature in the *Very important* category is twice as important as a feature in the second category, which in turn is twice as important as a feature in the last category. If this is not true, the weights are inaccurate or the feature is wrongly categorised. It may also indicate that more categories are needed.

The summarising process

When the importance of each attribute has been established, it is time to see how well each product scores. Other evaluations answer questions such as "Can formats other then HTML be indexed?" with a simple yes or no. We recommend against that type of binary scoring, extending Saracevic's (1995) argument to say that most answers may be graded to better portray the complex reality. Furthermore, a metric measure was preferred to words such as *excellent*, *good*, *fair*, or *poor*, which are otherwise often used (e.g. Chu and Rosenthal, 1996). When many aspects are considered it is difficult to calculate the average of 8 "g*ood*", 12 "f*air*", and 3 "*poor*". Therefore, we recommend that a 6-point scale should be used, where 5 means full support and 0 means no support.

We suggest that not only the grand total for each product should be calculated, but that summarising should be done both per importance category and functionality area. *Summarising both per importance category and functionality area adds a dimension to the evaluation. Not only is the overall winner identifiable, but also the particular characteristics for each product.* We argue that these characteristics are just as important as the grand total.

To summarise per importance category, first create a table such as table 6 below, and enter the items from the "Very important" category. Enter the score and calculate the sub-total, do the same for the other categories, and finally enter the sub-totals into the table as in the example in table 7. The product best suited for the needs of the organisation should now be easy to identify. Other potentially useful information can also be read from the table, for example that the second runner-up (Candidate #1) "wins" in the "important" category.

Table 6: Example of a "Very important" category table in a 4-candidate evaluation sheet. The weight for this category is set to 8 and each aspect can receive 1-5 points. The table shows both points and total (e.g. 5/40)

Question	Candidate #1	Candidate #2	Candidate #3	Candidate #4
Proximity search?	5/40	4/32	3/34	5/40
PDF support?	3/24	5/40	5/40	2/16
Frame support?	5/40	5/40	5/40	0/0
Natural language?	4/32	5/40	5/40	0/0
Numbers?	5/40	5/40	5/40	5/40
Custom search?	5/40	4/32	5/40	4/32
Custom result?	2/16	5/40	2/16	1/8
Real-time update?	5/40	5/40	5/40	0/0
Sum:	**272**	**304**	**280**	**136**

Table 7 below shows that Candidate #2 is the product best suited for the particular needs of the organisation in question. However, it is useful to analyse the data further by summarising per functionality area also. To do this, create a table such as table 8 below for each of the five functionality areas. Enter the weighted scores and calculate the sub-total for each area. Table 8 below shows an example of how this summary might look for the data gatherer category with a few fictitious attributes entered. It shows that Candidate #3 has the best data gatherer although Candidate #2 was the overall winner.

Table 7: The overall scoring table of the example evaluation sheet. Here, three importance categories are added to a grand total.

Total score	Candidate #1	Candidate #2	Candidate #3	Candidate #4
Very important	272	304	280	136
Important	96	88	80	68
Useful	64	64	60	52
Sum:	**432**	**456**	**420**	**256**

Summarising per functionality area and comparing these categories, as in table 9 below, help identify the strengths and weaknesses of each product. The possibility of being able to analyse the score according to both importance and function may prove useful should two or more products end up with the same overall score.

Table 8: Summarising per functionality area. The items from the three different importance categories, weighted 8, 4, and 2, respectively, are included. Both points and total (e.g. 5/40) are displayed.

Data gatherer	Candidate #1	Candidate #2	Candidate #3	Candidate #4
PDF support?	3/24	5/40	5/40	3/24
Frame support?	5/40	5/40	5/40	0/0
Robot exclusion	5/40	5/40	5/40	5/40
Image maps?	5/20	0/0	3/12	3/12
Domino?	0/0	5/20	5/20	3/12
Usenet?	5/20	5/20	4/16	3/12
Find directory?	5/20	0/0	5/20	0/0
URL check?	0/0	4/8	4/8	0/0
Password?	0/0	2/4	4/8	0/0
Sum:	**164**	**172**	**204**	**100**

Table 9: Adding all five functionality categories to a grand total. The column sums should be the same as in table 7.

Functionality area	Candidate #1	Candidate #2	Candidate #3	Candidate #4
Platform, etc.	30	28	30	20
Data gatherer	164	172	204	100
Index	160	164	134	90
Search features	50	60	28	24
Operations, etc.	28	32	24	22
Sum:	**432**	**456**	**420**	**256**

Install and test

Each of the two organisations studied installed three products and noticed several important differences between the answers given by the vendors and the experienced results. This need not result from the vendors being untrustworthy, but more likely due to different interpretations or levels of experience. What the vendor might consider to be "good documentation", the end-user might rate as "poor", a difference often undetected until the documentation is made available for evaluation. Knowing what to verify and where to check for strengths and weaknesses also made testing more effective. Both companies also conducted precision tests after having installed the most promising candidates.

Conclusions

Running an intranet search engine is different from merely using a public search service, since otherwise ignored aspects such as choice of platform, operability, maintenance, and support become important. Intranet search tools should thus be evaluated differently – a fact that seems to have gone undetected. We have collected and listed a set of features that should not to be seen as a final set, but rather as a starting point, to which new findings should be added. We believe that by identifying as many aspects as possible, it is less likely that something important will be missed. Our main point, though, is that you

do not know which of these features are important and which are not until you are about to implement a search engine yourself. We therefore recommend that the focus should be on the questions rather than on soon outdated answers. The individual items are then *grouped* into categories, *ranked* and *weighted* according to importance at the site actually *doing* the evaluation.

Once the necessary answers have been collected, the overall winner is easily identified, at the same time displaying both its strong and weak sides. It is also possible to account for any decisions to discard a certain product since it is very easy to show that the product in question had a low overall score, or that it scored poorly in a particular functionality area. As business evolves over time and products keep improving, re-evaluation of the decision might become necessary. To have documented that a certain product was discarded due to lack of a specific feature may be useful if it turns out that the feature now is available, or that the feature is no longer required.

Though none of the individual ingredients used in this recipe is in itself new or revolutionary, they have never before been combined and used in this particular context. Our approach is in that sense both new and useful. It may be argued that manually assigning scores to features will bias the evaluation and open it to subjective opinions. We realise that this is an issue for criticism, but still believe that this methodology helps reduce those risks. Before having summarised, it is difficult to know how the individual products are doing, and since the concentration is targeted on a single feature at the time, there is less risk of *unintentionally* favouring any particular product. If one truly *wants* to favour a product, it can be done using any methodology.

Acknowledgements

We would like to thank staff and management from the two companies for assisting us during this research, and Mark Parkins at Corporate Express for evaluating the methodology and providing valuable comments.

References

Chu, H., and Rosenthal, M. "Search Engine for the World Wide Web: A Comparative Study and Evaluation Methodology", In *Proceedings of the 1996 ASIS*, October 19-24, 1996.

Furnas, G., Landauer, T., Gomez, L., and Dumais, S. "The Vocabulary Problem in Human-Systems Communication", *Communications of the ACM*, Vol. 30, No. 11, pp 964-971, November 1987.

Goldschmidt, R., Mokotoff, C., and Silverman, M. "Evaluation Of WWW Search Engines For NIH Implementation", National Institutes of Health, Bethesda, MD, USA, January 30, 1997. Available at *http://bigblue.od.nih.gov/websearch/report.htm* [Feb. 1999].

Hearst, M. A. "Interfaces for Searching the Web", Special Report Article, *Scientific American*, #3 1997.

Koster, M. "A Standard for Robot Exclusion", consensus on the robots mailing list (robots-request@nexor.co.uk) as of 30 June 1994.
Available at *http://info.webcrawler.com/mak/projects/robots/norobots.html* [June 1999]

Lange, A. "Sorting through Search Engines", *Web Techniques Magazine*, Volume 2, Issue 2, June 1997

Leighton, H. V. and Srivastava, J. "Precision among World Wide Web Search Services (Search Engines): AltaVista, Excite, Hotbot, Infoseek, Lycos", June 1997.

Available at *http://www.winona.msus.edu/library/webind2/webind2.htm* [June 1999]
Nance, B. "Internal Search Engines Get You Where You Want To Go", *Network Computing Online*, October 8, 1997.
Notess, G. R. "Searching the World-Wide Web: Lycos, WebCrawler and More", *Online*, July 1995.
Nielsen, J. "User Interface Directions for the Web", *Communications of the ACM*, Vol. 42, No. 1, pp 65-72, January 1999.
Pinkerton, B. "Finding What People Want: Experiences with the WebCrawler", In *Proceedings of the Second International World Wide Web Conference*, Chicago, Illinois, USA, July 1994.
Raghavan, V., Bollman, P., and Jung, G. S. "A critical investigation of recall and precision as measures of retrieval system performance", *Communication of the ACM*, Vol. 7, No. 3, pp 205-229, July 1989.
Rao, R., Pedersen, J., Hearst, M., Mackinlay, J., Card, S., Masinter, L., Halvorsen, P.-K., and Robertson, G. "Rich interaction in the digital library", *Communications of the ACM*, Vol. 38, No. 4, pp 29-39, April 1995.
Saracevic, T. "Evaluation of evaluation in information retrieval", In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Special issue of SIGIR Forum, pp 138-146, 1995.
Shneiderman, B., Byrd, D., and Croft, W. B. "Sorting Out Searching. A User-Interface Framework for Text Searches", *Communications of the ACM*, Vol. 41, No. 4, pp 95-98, April 1998.
Slot, M. "Web Matrix: What's the Difference? Some answers about Search Engines and Subject Catalogs", 1995-96.
Available at *http://www.ambrosiasw.com/~fprefect/matrix/answers.html* [June 1999].
Turtle, H. R. "Natural Language vs. Boolean query evaluation: A comparison of retrieval performance", In *Proceedings of SIGIR '94*, pp 212-220, Dublin, Ireland, Springer Verlag, 1994.