Blossom

What makes a good website search engine?

In a nutshell, a good search engine should return everything relevant to a query, but nothing that isn't relevant. It should be clear from the search engine results why each page was returned and whether the page really is relevant. If the search engine doesn't find anything for a query, it should suggest changes to the query that will find something. Alternatively, if a search engine finds too much, it should suggest changes that will narrow the search.

Read on for more details about each of these points.

All the facts, but just the facts

The most basic measures for judging the results of a search engine are known as recall and precision. *Recall* assesses whether the engine found all the relevant pages. *Precision* assesses how many of the pages returned were actually relevant. Good search engines have both high recall and high precision; they return most of the relevant pages without returning pages that aren't relevant.

As you might guess, increasing recall usually decreases precision. Interpreting a query more broadly returns more pages, but some of those pages won't be relevant to the searcher. A simple illustration of this is to look at three Boolean operators: OR, AND, and NEAR. For a simple query, a search engine interprets the search terms as being connected by a Boolean operator.[1] For an example, let's use the query "good search engine". A search engine that uses the OR operator would treat the query as

> good OR search OR engine

The results would include all pages that contain any of the words "good", "search", or "engine". It is easy to see that a search using OR will return many irrelevant pages, particularly in this case because we would expect the word "good" to appear on many pages that don't mention search engines.

[1] Although most search engines, these days, implement a more complex search than a simple Boolean, they can still be categorized as using one of the three (extended) Boolean operators.

Some search engines use the AND operator:

good AND search AND engine

The results of an AND search are more precise than using OR because all of the search terms must appear on a page for it to be chosen. The recall of AND is lower than OR because the results are a subset of OR search; any page returned by AND will also be returned by OR. Even so, an AND search can still include many irrelevant pages because although the words all appear somewhere on the page, they might appear in wholly separate contexts, such as “After a long *search* I got a *good* deal on my car … It is a 2008 Honda Accord with a hybrid *engine*.”

The NEAR operator selects pages where the search terms appear close to one another, thus

good NEAR search NEAR engine

finds pages where “good”, “search”, and “engine” appear nearby, perhaps in the same sentence. Search using NEAR is called *proximity* search. Proximity search has the highest precision, but also the lowest recall because its results are a subset of AND. To counter low recall, a good proximity search engine will offer suggestions to guide the searcher.

Why these results?

A key attribute of good search engines is transparency, that is, they show clearly why each returned page was chosen. Page “snippets”, those extracts from a page that show the search terms in context, contribute significantly to transparency. Search engines that just return page titles or general descriptions of a page usually don’t make it clear why the page matches a particular query.

If the snippet is large enough, it can also help the searcher decide whether the page really is relevant. Proximity search has an advantage here because all of the query terms should appear in the snippet. For AND and OR searches, a small snippet may only have some of the query terms.

What next to search?

Sometimes when searching a website a query either comes up empty or has too many results. A good search engine should help modify the query to get better results.

Consider the case of empty results. For an OR search that means none of the query terms appears anywhere. In this case it’s hard to know what to suggest other than “try different terms”. (Of course, an OR search rarely comes up empty; usually the problem is too many results, so see the next paragraph.) For an AND or NEAR search the follow-up is easier. Empty results just mean that all the query terms do not appear on the same page, so dropping one or more terms might give more results. The best engines suggest which terms to drop.

What about the case of too many results? Again an OR search has a difficult job. To get fewer results with OR, the search engine should suggest *dropping* terms (because with OR, each term potentially adds pages to the results.). But dropping terms intuitively makes the search *less*, rather than more, specific. That is, "search engine" is less specific than "good search engine".

For an AND or NEAR search, adding terms does make a search more specific, but what terms should be added? The best search engines suggest terms to add based on how the terms will affect the search results.

Give me control!

A significant benefit of having your own search engine is that you can control its behavior, such as the matching of pages to queries, the ordering of pages in the results, and the overall appearance of the results. Also, to respond quickly, search engines answer queries by consulting a search index that is created offline. Having an up-to-date index is important for returning accurate results.

Meta information, like keywords and page descriptions, provide one way to influence the search results. Unfortunately, Web search engines often ignore meta information because it is a common source of search engine spam.[2] By contrast, you probably trust the sites in your search index, so a good site search engine should use meta descriptions and keywords to improve the search results.

But spare me the headaches

Search engines actually consist of a suite of software for creating, configuring, and searching. Like any complex software system, to get the best results requires informed administration and maintenance. A key decision is whether to install the search software locally or utilize search as a service. Below are three issues to consider:

- Running a search engine is similar to running a Web server. Depending on the size of the search index, building the index and serving queries can require significant computing resources. You will need to keep these requirements in mind as you choose your hardware.
- Because of past well-publicized security flaws, search engines have become a popular avenue of attack by hackers. Taking advantage of program bugs and insecure scripts, hackers have been able to get full access to computer systems running search. If the system has access to private data, then the search engine must be treated as a potential security breach.
- Neither the capabilities of the search engine nor the environment in which it runs are fixed. Search software evolves to improve search results, to take advantage of new Web technology, and to thwart new types of attack. As for other key server software, you need to keep search software up to date.

[2] Search engine spam is an attempt by unscrupulous webmasters to trick a search engine into selecting pages for topics unrelated to the actual page content.

It is for these reasons of performance, security, and maintenance that “software as a service” (SaaS) is good model for search. With SaaS, rather than installing software on your systems, all software and data are kept on a server run by a service. The service company is responsible for allocating enough hardware resources, protecting the search engine and data, and updating the software. In the best case, once you’ve connected your search form to the search service, your job is done.