Search Results Clustering Versus Query Refinement

Query Refinement

In query refinement approaches, a user types a search query and then is shown search results plus possible new search queries, which when clicked on issue a new query whose results replace the current results seen by the user. The goal is to encourage users who are dissatisfied with the top search results to query again rather than leave the site.

- "Query refinement can be seen as a mechanism that recommends query modifications to reduce false positives."

- "Query refinement is the incremental process of transforming a query into a new query that more accurately reflects the user's information need."

These two quotes from academic research articles reveal the intent behind query refinement: to issue an improved search query that supplants or even lengthens the user's original query. The original query is held to be flawed because it returns too much irrelevant information. The original search results are discarded and brand new ones that are supposedly more accurate are shown instead.

Problems with Query Refinement

There are problems with query refinement, both in principle and in the way it is practiced.

1. Query refinement is **discontinuous**: when the user clicks on a refinement, the information on the screen totally changes, forcing the user, who may just be browsing, to absorb new information and later to resort to backtracking to retrieve the old. Discontinuity leads to an inferior user experience.
2. Traditionally, query refinement quality has not been very good. As a result, either little or **inferior screen space** is devoted to it, so it doesn't get clicked on much because the user's eye is not drawn there. Typically query refinements are placed at the bottom or the right of the page, which are hardly the places to put top-quality information.
3. Access to the **search index** is needed. This means that implementing query refinement in meta-search mode, where an external search engine is called, is problematic. For example, an external paid-listings provider (Overture, Google, FindWhat) might be called in meta-search mode. Since these external sources aren't pre-processed, the commercial themes reflected there may be absent from the refinements, undermining the chances of seeing click-throughs to paid listings and hence revenue.
4. Since query refinement involves significant pre-processing, the refinements will **lag** the top search results. For example, none of the 12 Prisma query refinements suggested by a 9/23/03 AltaVista search on Arnold Schwarzenegger turned up any hint of his campaign for governor, even though two of the top 10 AltaVista search results mention *governor*.
5. There is no link between the query refinement options and the current search results on the screen. Therefore, one cannot easily implement advanced user features which depend on navigating the information **bottom-up**, i.e., from individual search results to

groupings (clusters) of them. Vivisimo.com's S*how-in-Clusters* feature is an example of bottom-up navigation.

6. If the **query is too long**, then in some implementations no refinements are suggested, even though the returned results are numerous enough for the user to need help in making sense of them. For example, AltaVista Prisma offers no further refinements after the successive queries Cancer *then* Colon Cancer *then* Diagnosis, even though a user clearly can benefit from more help in sorting through the final 35,285 results.
7. Because refining a query issues a new search, the user **cannot easily compare** two candidate refinements, short of opening up multiple windows on the screen.

On-The-Fly Clustering of Search Results

In the on-the-fly clustering approach, the idea is to capture a large sample of the top-ranked search results, cluster them into meaningful folders that extract the major themes, and let the user explore the folders. There is no presumption that the user's query was somehow poorly expressed. On the contrary, broad searches, which are traditionally prime targets for query refinement, are welcomed.

Most of the earlier problems are absent from clustering approaches, when executed well.

1. Clustering is continuous: when the user expands a folder or views the search results that it contains, localized changes occur on the screen. The larger context is kept.
2. Clustering can lead to poor folder descriptions or to good ones. The best approach works well and provides the confidence that lets search providers allocate a fixed, prominent screen space to the folders. Empirical data show that users click on folders at a high rate.
3. No access to the search index is needed. On-the-fly clustering allows combining search results from anywhere (e.g., paid listings) with equal quality, as long as the search results possess decent titles and snippets.
4. Clustering uncovers themes that are as fresh as the top-ranked results. No missing Arnold Schwarzenegger's run for governor, or any other recent events that are reflected in the top search results.
5. The folder themes are linked to specific search results, which enables advanced bottom-up navigation features such as Vivisimo.com's *Show in Clusters*, or the more familiar Find in Clusters (search within the top results), both of which highlight in yellow the folders that contain the selected or matching search results..
6. Meaningful folders are always extracted no matter how long the query is, as long as 50+ search results are returned.
7. Folders can be expanded independently within the same window, so the material under two or more folders can be easily compared and contrasted.

The disadvantage of clustering is that it requires a minimum of 100-200 search results in best practice. This was a problem several years ago, due to slower computers, more expensive servers and bandwidth, immature business models for web search, and sparser coverage of search keywords by advertisers.

These obstacles are largely gone, partly because of the rise in paid-listings search and the monetization opportunities that clever implementations of clustering enable. Details on the latter are outside the scope of this white paper.