

Confluence: Enhancing Contextual Desktop Search

Karl Gyllstrom, Craig Soules, Alistair Veitch

karl@cs.unc.edu, craig.soules@hp.com, alistair.veitch@hp.com

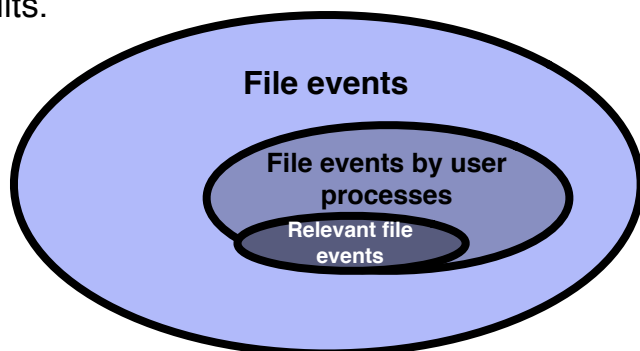
Problems With Desktop File Search

No hyperlinks means weak structure. Search can't benefit from PageRank, HITS, and other structural search methods which perform well on the web.

There is little information to enable reasoning about file relationships and importance.

New Challenges

File information is obscured by noise. Virus checkers, music players, mail, and other applications access files in the background. This leads to many **false** relationships that degrade the quality of search results.



Evaluation

We deployed **Confluence** with 4 users over a 3-6 week period. After the period, users identified a small set of files, where each selected file pertained to a different task (e.g. writing a paper, creating a presentation). These files were individually used as *queries* for which contextually related files were found.

We evaluated Confluence by **recall**; in this case, the ability of Confluence to identify other files used within the same task of the user-selected file. In total we evaluated 31 queries.

Applying Temporal Locality

Confluence is an extension to **Connections**[1], a desktop search tool which identifies *contextual file relationships* by similarity in their *access patterns*. Contextual relationships can be used to augment traditional search methods with additional, conceptually related files that do not match the text query.

For example, if documents A and B are frequently accessed at similar points in time, this suggests a task commonality. Searches that return "A" now return "B" as well.

Approach

Confluence records *window focus* events within the *GUI*, which are generated each time the user activates a different application window. These events are used to infer *task*.

File events which are not *causally* related (i.e. are not owned by the process or subprocess that caused the event) to a recently focused window are ignored. This allows **Confluence** to isolate file events which are more likely to be from *direct user activity*.

Results

We achieved significant gains in recall, comparing the UI-enhanced method to the pure file-based approach.

Within the top **30** files identified as contextually related by the UI-enhanced method, **%45** of the other files in the task were identified. For the pure file-based approach, **%20** were identified.

[1] C. A. N. Soules and G. R. Ganger. *Connections: using context to enhance file search*. In Proc. of SOSP '05, pages 119–132, New York, NY, USA, 2005. ACM Press.

[2] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. *Stuff I've seen: a system for personal information retrieval and re-use*. In Proc. of SIGIR '03, pages 72–79, New York, NY, USA, 2003. ACM Press.

[3] D. Karger, K. Bakshi, D. Huynh, D. Quan, and V. Sinha. *Haystack: A General Purpose Information Management Tool for End Users of Semistructured Data*. In CIDR '05, pages 13–26, 2005.

