Retaining Queries in Noflail™ Search

Francisco Corella
December 2008

1. Introduction

Noflail™ Search is a Web search facility developed by Pomcor, available at noflail.com, which interfaces to the Yahoo search engine through a Yahoo Web API (currently the Yahoo Boss API, complemented by other Yahoo Web APIs as needed). Version 1 of Noflail Search, described in a Pomcor white paper, helped the user with difficult search problems by anticipating follow-up queries to an original query entered by the user, issuing them immediately and simultaneously as additional queries, and allowing the user to browse the result sets of the original query and the additional queries simultaneously.

To accomplish this, Noflail Search Version 1 placed the original and additional queries in a left panel and showed their result sets in a center panel. With a single click on a query, the user could cause the center panel to switch to the result set of that query without delay. A page menu at the bottom of the center panel allowed the user to browse the result set.

This paper describes two major new features of Noflail Search, introduced in Version 2. While in Version 1 all queries were cleared from the left panel when the user issued a new query, the user can now keep some of them, and browse the result sets of old and new queries simultaneously. We refer to this feature as *query retention*. Furthermore, queries listed in the left panel are saved to a persistent store and are not lost when the user exits Noflail Search, or closes the browser, or turns off the machine. We refer to this feature as *query persistence*. (The terms *query retention* and *query persistence* have similar meanings in English but we use them here to refer to two different features for explanatory purposes.)

Query retention was originally conceived as an improvement on a mechanism for exploring the subquery space of a given query. When combined with query persistence, however, it becomes an important feature in its own right.

2. Retaining queries

To allow the user to keep old queries in the left panel when a new query is issued, Noflail Search Version 2 places a checkbox next to each query. The left panel is now a table with one row per query and three columns, as shown in Figure 1. For each query, the first column contains the checkbox of the query, the second column contains the text of the query, and the third the number of results of the query.

When the user issues a new query, the new query and any additional queries are inserted at the top of the left panel, before any older queries that may have been retained by the user. The newly inserted queries all have checkmarks in their checkboxes. A checked query is interpreted as having been selected for deletion. Deletion of checked queries

occurs when the user clicks on a *Delete* link visible at the top of the left panel in Figure 1. It also occurs automatically when the user issues another query.

If the user does not remove any checkmarks, all the inserted queries go away when the user issues another query. On the other hand, if the user removes one or more checkmarks, the corresponding queries *do not go away* when the user issues another query; this is the *query retention* feature. Thus the query retention feature has two properties:

1. It carries no cost to the user, since the user does not have to take any action to avoid retention.
2. It can be used with minimal effort, since the user can retain a query with a single click that removes its checkmark.

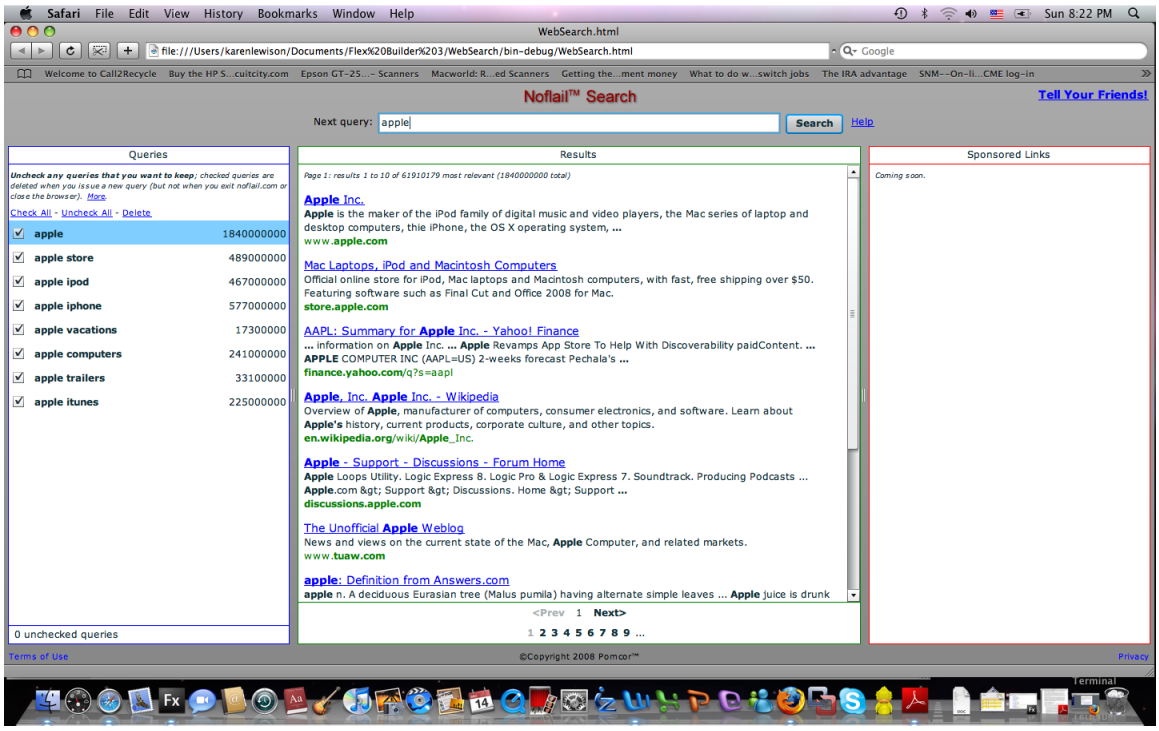

Figure 1

3. Exploring the subquery space of a query

Noflail Search Version 1 had a mechanism for exploring the subquery space of certain queries. When the user issued a query and the Yahoo Web API suggested no related queries, Noflail Search used subqueries of the user's query as additional queries. In the case where the user's query had results (rather than being a *zero-result* query), the additional queries were the subqueries obtained from the user's original query by

removing exactly one of the search terms. Thus if the user's query had *N* terms, the additional queries were the subqueries with *N* – 1 terms. To run a subquery with fewer than *N* – 1 terms, without typing it in, the user could double-click on a subquery with *N* – 1 terms, or drag it from the left panel to the query box. This had the effect of issuing the subquery as the next query and producing its own subqueries (assuming that the Yahoo Web API suggested no related queries), which had N – 2 terms; and the process could be repeated.

The *query retention* feature was originally conceived as an improvement on this mechanism. In Version 1, when a subquery with *N* – 1 terms was issued as a new query, the original query and the other subqueries with *N* – 1 terms were wiped out from the left panel. In Version 2, the user can remove the checkmarks of any of those queries (those that look promising for the search problem at hand) causing them to be retained when the new query is used. The user can thus accumulate promising queries with any number of terms in the left panel, and browse the result sets of all of them simultaneously.

4. Query persistence

Besides keeping old queries when a new query is issued, Noflail Search Version 2 keeps all left panel queries upon exit. Queries are saved to persistent storage, and are recovered from persistent storage when the user visits *noflail.com* again using the same browser running on the same computer. This is the *query persistence* feature.

All queries are saved to persistent storage, whether checked or not. Saved checked queries will be deleted when the user visits *noflail.com* again and issues a query, if their checkmarks have not been removed before the query is issued.

Each query saved to persistent storage is annotated with the page number of the last page that the user visited in its result set (using the page menu at the bottom of the center panel). When the query is recovered from persistent storage and placed again in the left panel, and the user clicks on the query, the center panel goes directly to that page.

5. Combination of retention and persistence

Query retention and query persistence are two independent features which, when combined together, provide major benefits. These benefits go well beyond the improvement of the subquery space exploration mechanism that was the original motivation for query retention.

Queries kept in the left panel do not have to be related. The left panel can be used to build a collection of useful queries that can be used again in the future. Saving and reusing a query may be useful for two different reasons. First, a query may yield many useful results, and it is then easier to save the query than to visit and bookmark each of the results. Second, the user may want to issue a query repeatedly if the result set of the query changes over time.

It should be noted that it is possible for a user of a traditional search engine, such as Google or Yahoo, to save queries. To save a query, the user can issue the query and then bookmark the page of results produced in response. However the left-panel facility of Noflail Search is superior to query bookmarking for several reasons:

1. Query bookmarks are, by default, interspersed with other bookmarks, whereas the left panel contains only queries.
2. The left panel shows the number of results of each saved query, whereas bookmarks do not.
3. Queries saved in the left panel are immediately visible and accessible when the user visits noflail.com. Noflail Search has an initial display without panels; but it transitions immediately to a subsequent display with panels if it finds saved queries in the persistent store. The transition is accompanied by a visual effect that makes it clear to the user that the transition has taken place automatically.

Noflail Search is implemented on the Adobe Flex platform, and the query persistence feature takes advantage of this. It uses a Flex *local shared object*, informally known as a *Flash cookie*, as persistent storage. A Flash cookie provides 100 KB of persistence storage by default, enough for storing hundreds of cookies.

5. Remarks

5.1 In Noflail Search Version 1, queries were inserted into the left panel only after their results were available. In Version 2, in most cases, a query is inserted as soon as it is issued, and the number of results it produces is added later, when the response to the query is received from the Yahoo Web API. This is an improvement, because it allows the user to inspect the list of additional queries derived from a user's original query without waiting for their results to arrive. This cannot be done, however, in the case where the original query has zero results, since the additional queries in this case are determined as their results arrive.

5.2 As another improvement in Version 2, the user is now allowed to reorder the queries in the left panel by dragging them with the mouse. Without any reordering, queries appear in chronological order, with the most recently inserted queries at the top of the list. Reordering allows the user to move old-but-important queries towards the top of the list, where they are accessible without scrolling.

5.3 The number of queries kept in the left panel could have a fixed limit, or a limit based on the amount of space they occupy in persistent storage. A fixed limit is used in Noflail Search Version 2 for the sake of simplicity. The limit applies to the number of unchecked queries (queries without a checkmark), because unchecking is performed by the user, whereas the user does not control the number of additional queries that are inserted automatically when the user issues an original query. Queries are inserted with checkmarks, so query insertion does not affect the limit if the limit applies to unchecked queries. When the limit has been reached, the user can continue to uncheck queries; but for every query that the user unchecks, another query is checked automatically to stay within the limit. The query that is checked is the last unchecked query in the list other than the one checked by the user.

5.4 Noflail Search is implemented on the Adobe Flex platform, and takes advantage of the features offered by Flex, such as transition effects and 100 KB of default persistent storage. However, it could also be implemented on other platforms, such as Silverlight or Javascript. If implemented in Javascript, an ordinary cookie could be used for persistent

storage. The amount of storage provided by an ordinary cookie may be as little as 4 KB (depending on the browser), but this may be enough for about 100 queries (depending on the queries). Noflail Search could even be implemented in pure HTML with no client-side scripting, again using an ordinary cookie for storage; the Noflail Search logic would then be implemented on Web servers. Requests to the Yahoo Web API would then be sent by those Web servers, whereas in the Flex implementation they are sent directly from the client machine.

5.5 A search engine provider, such as Yahoo, could move part of the Noflail-Search functionality from the client machine to the search engine itself. The search engine could take care of determining the additional queries and would could send the list of additional queries and their results to the client without the need for additional requests from the client.

5.6 The left-panel queries could be stored on the server-side rather than on the client side. To enable this, the user would register to obtain a user ID and password or other login credentials. The user could then have to choice to log in and use Noflail Search with access to his or her query repository, or use Noflail Search anonymously without access to the query repository.

5.7 Noflail Search could be implemented as a client-resident application rather than a Web application. This would be particularly easy to do with the existing Flex implementation, since the same code that now runs on the Flash plug-in could run on Adobe Air. To enable this, the user would have to agree to install Noflail Search and Adobe Air on his or her computer. When implemented as a client-resident application, Noflail Search could save the left panel queries to a file in the file system of the client machine, each with its number of results (if available) and a flag indicating whether it is checked or not.

5.8 In Noflail Search Version 2, queries kept in the left panel are structured as a simple list. A future version of Noflail Search may allow the user to organize the queries in a hierarchical structure of folders and subfolders. The *Tree class* of Flex would facilitate the implementation.

5.9 Version 1 of Noflail Search used the browser's History feature. Issuing a query changed the URL in the query box of the browser, and the user could use the back and forward arrows of the browser to navigate the URL history. This feature has been removed in Version 2, because it is awkward, if at all possible (depending on the number of left panel queries), to encapsulate the state of the left panel in a URL. A future version may provide *Undo* and *Redo* commands to allow the user to navigate the history of interactions with Noflail Search.