"Google* by Reformulation": Modeling Search as Successive Refinement

Stephen Davies[1], Serdar Badem[1], Michael D. Williams[2], Roger King[1]
[1]*University of Colorado, Boulder {stephen.davies, serdar.badem, roger.king}@colorado.edu*
[2]*Caring Family, LLC mwilliams@caringfamily.com*

Abstract

Humans naturally employ a process of iterative refinement when they search for information, clarifying their intentions and adjusting their goals in response to what they encounter. We propose to develop tools specifically to facilitate this process, and hypothesize that they will help users converge on their desired information more quickly. This paper presents a layered extension to the Google Web search interface that illustrates the so-called "retrieval by reformulation" paradigm. The tool will serve as a launching point to a wider field of inquiry into user interaction with large information spaces.

1. Introduction

The search for new information is inevitably an iterative process. True, there are times when we begin with in-depth knowledge of a database's contents, and merely want to retrieve a specific record that we already know exists. But this is only a matter of maintaining our existing knowledge, not accumulating new knowledge. As soon as we begin to explore unfamiliar territory, we enter a world of trial and error. We usually start with only a vague notion of what it is we're looking for, and our goal evolves as we gradually learn what the information space contains. Only after a series of failed attempts and discoveries do we find out what information is actually available, and how we need to ask for it.

Nowhere is this more true than with the World Wide Web, the world's largest and most diverse database. It is so enormous that most often we don't know exactly what it is we're looking for until we find it. Our initial set of search terms is like a shotgun fired into cyberspace, reeling in a motley assortment of pages that tell us as much about what we're *not* interested in as about what we are. Over time, we filter and assimilate the meaningful information. We pursue leads and dodge dead ends, tweaking and refining our search repeatedly until perhaps at last we recognize in some distant page the information we originally set out to find. The process is tedious and error-prone, and up to the user to carry out entirely on his own.

Though the tools don't support it, a user will nevertheless manually execute the only possible strategy to achieve his goal: a process of iterative refinement. Through interactive discovery, the user, search engine, and world of information jointly impinge on one another to blaze a path through the forest towards the desired knowledge. The user continually refines his questions in response to what he encounters. The signposts help him to clarify his ideas along the way, and may even cause him to change his mind about where he wants to go.

This is in fact representative of nearly everything a human being does. We do not begin with omniscient knowledge, but rather interact and experiment with our environment, learning over time the relevant cause-effect relationships and how to get done what we need to. It is nearly impossible to start in a vacuum and know exactly what we want and how to ask for it. Yet with their passive, stateless nature, most modern search engines are asking us to do just that. Any iterative refinement is left entirely up to us to manufacture.

2. Retrieval by reformulation

We believe that the paradigm of *retrieval by reformulation* can offer an advantage to the stateless interface. The idea was originally developed over two decades ago and was based on psychological theories of human remembering. Its purpose is to aid users in formulating effective queries by explicitly guiding a process very much like the one described above. Two essential characteristics of the paradigm are:

- Query by partial description. Rather than specifying the location of a desired object or using a query language such as SQL, the user initiates a query by *describing* the object he is seeking. The user thus does not think primarily in terms of attributes, hyperlinks, or relations, but in terms of

* Google is a trademark of Google, inc.

whole instances. With a description of the user's goal, the system has a starting point from which to begin the search process.

- Critique of example instances. This is the core of the paradigm. The search process is one of successive refinement whereby the user looks at actual objects from the database and provides feedback about what he likes or doesn't like about them. This allows the system to focus the search and rapidly converge on the most relevant data.

Retrieval by reformulation has other advantages besides helping searches converge quickly. It also makes clear what to look for by repeatedly showing actual example data – this helps users see what kind of data is present, understand what their options are, and better crystallize their thoughts and expectations.

Essentially, retrieval by reformulation can be seen as making explicit the procedure that users instinctively undergo when exploring an unknown realm. Humans will naturally respond to the data they encounter, and adjust their search strategies in light of it. But without tool support, this process is bound to be erratic and haphazard, with valuable information overlooked, misapplied, or stored only in the user's mind where the search engine cannot make use of it. A retrieval by reformulation tool aims to bring this methodology to the forefront of the user's consciousness, encouraging him to provide the most comprehensive feedback possible in order to guide the search.

One of the earliest experimental tools was Williams' and Tou's RABBIT database interface[10]. It implemented the paradigm for structured, relational data, but we believe many of the ideas are applicable to natural language text as well.

3. The Web: Google by reformulation

To illustrate these ideas for the realm of Web information retrieval, we developed a prototype plug-in extension to Microsoft Internet Explorer called "SearchPlus." It uses Browser Helper Objects (BHO) technology to integrate a dynamic link library (DLL) directly into the browser. This allows us to intercept API calls, detect and respond to user interface actions, modify popup menus, add toolbars, etc. We can thus take advantage of the functionality already provided by the browser, and simply add support for the reformulation process.

The fundamental idea is that instead of forcing the user to manually refine and resubmit a succession of queries, this process is incorporated directly into the interface. The user begins by initiating an ordinary Google search with a set of terms expressing the content she desires. This is identical to how the world uses search engines today, except that the list of terms will not be static, but will grow as the search process ensues. The top ten hits from this search are displayed in the left pane. (See figure 1.) Choosing a link displays the corresponding page in its entirety in the right pane. It is at this point that the user has a chance to express feedback to the application. Upon examining the chosen document, she will typically render one of three judgments:

1. The page contains exactly the information the user was looking for, and no further searching is required. We call this a *terminal page* in the iterative search process.
2. The page does not contain enough information to terminate the search, but it is "on the right track." The user can discern that the semantic content of the page is closer to her mental target than the previous iteration, and hence the search is converging. She indicates this by identifying additional words in the selected document that are particularly suggestive of the desired content. In our application, this amounts to highlighting words with the mouse and selecting the "include keywords" function from the pop-up menu. The application simply adds these words to the current list of Google search terms that it maintains.
3. The page is diverging from the desired content: it is *not* "on the right track." Although all of the user's search terms were in fact present in the document, the perceived semantics of the page are quite different from what she has in mind. Here, she can use the mouse to highlight the words that indicate diverging content, and select the "exclude keywords" function from the pop-up menu. The application adds these words to the current list of search terms, but precedes each with a minus sign ("–") so that Google will *disallow* pages that contain those terms.

The goal in any search process, of course, is to quickly reach judgment number 1. When either 2 or 3 occurs, this indicates that the search process requires at least one more iteration in order to terminate. When the user has finished including or excluding various terms from the page, she selects the "Search in Google" popup menu option to start another iteration. The application simply feeds its list of current search terms (some of which may be preceded by a "–" if any previous iteration resulted in judgment 3) back to Google, which results in a new list of hits in the left-hand pane. The process continues in this fashion until either the search terminates, or the user abandons it because the perceived value of continuing is too small.

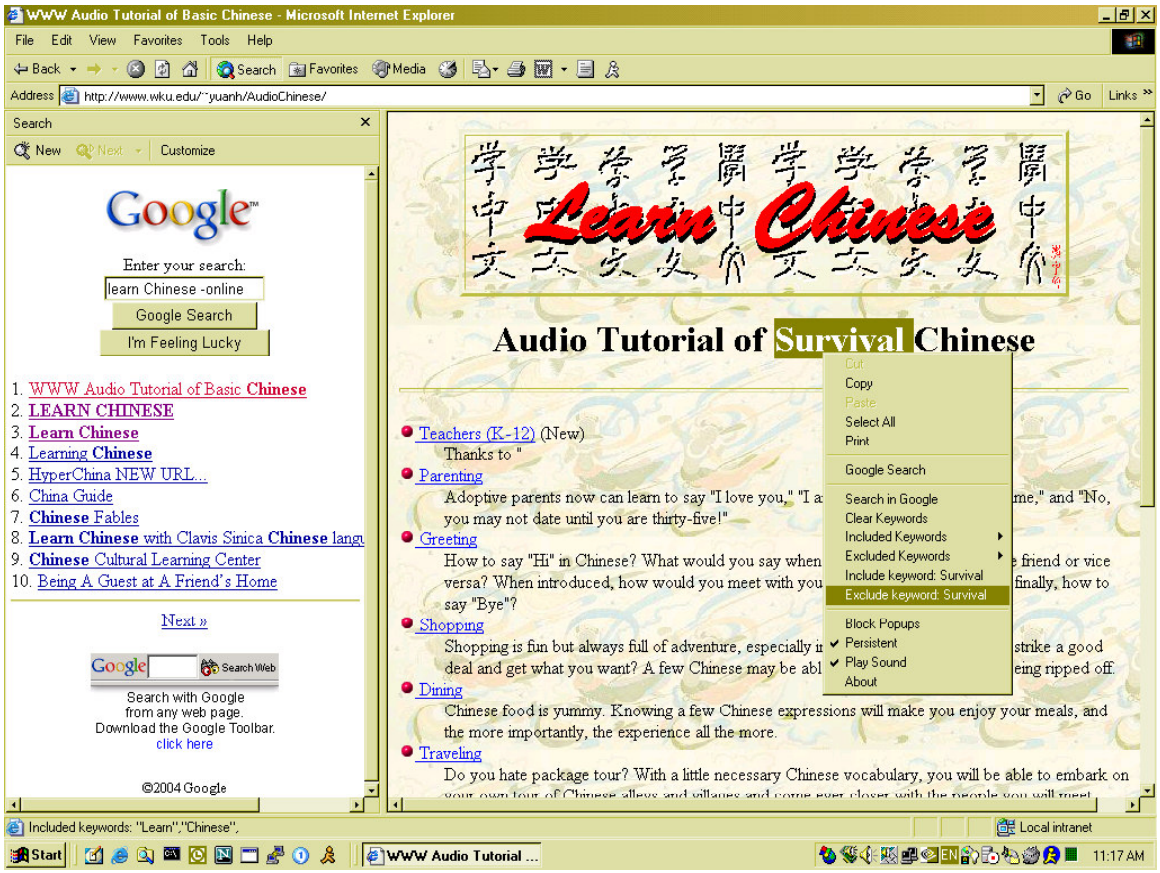

Figure 1. The "SearchPlus" Internet Explorer plug-in. At each step of an iterative search, the user chooses a page from Google's hit list and examines it in the right-hand pane. Based on their semantic interpretation of the content, the user refines their search by choosing various terms to either explicitly include or exclude from future searches. These terms are then included in the search term list and fed back to Google for the next iteration. In this way, the user can use corrective action to rapidly converge on their desired goal.

3.1 An example session

As an example of the system in action, we present the following scenario which we actually carried out using the tool. Suppose a businessman acquires the need to speak a new language fluently. Perhaps he is a westerner, and an opportunity has arisen for him to partner with an organization in China. He thus approaches the Web to find out what resources are available to help him learn to converse in Chinese. His initial Google search might simply be "learn Chinese." (This is certainly not atypical considering that the average number of terms in an Internet query is just over two.[3]) Using our system, then, here is a typical search scenario, broken down into iterations:

1. Google's results page with the top ten hits for the query "learn Chinese" is displayed in the left-hand pane. The businessman clicks on the first link, called "Learning Chinese Online Page" (the actual URL is http://www.csulb.edu/~txie/online.htm), and skims the contents. Right away he disregards the page, because he finds that it pertains to *online* language courses: this user has engaged in this sort of activity in the past and found that he much prefers face-to-face contact. To indicate this to the application, then, he highlights the word "online" within the page, and selects "exclude keyword" from the popup menu. Then, he selects "Search in Google" to spawn a new iteration. The system launches a new Google query with the terms "learn Chinese –online," and replaces the left-hand pane's contents with the new hit list.
2. The user browses the new hit list and elects to explore "WWW Audio Tutorial of Basic Chinese" (at http://www.wku.edu/~yuanh/AudioChinese.) After a moment's thought he discounts this page, too, since it seems to deal with only a handful of common phrases, rather than full-scale fluency in the language. Indeed, the page is subtitled "Audio Tutorial of Survival Chinese," so the user highlights the word "Survival" and again chooses "exclude keyword" to indicate that this term is *not*

indicative of the content he is looking for. (Figure 1 captures this instant of the search process.) He then requests a new search iteration, which the system performs on the basis of the terms "learn Chinese –online –survival."

3. The search now begins to converge. On the next hit list, the user finds a page called simply "Learn Chinese" (at http://www.ucalgary.ca/~chud/lchinese) that describes a continuing education course. Upon reflection, he realizes that this is indeed the sort of thing he is seeking: a noncredit, face-to-face instructional experience rather than a book or software tool. The page does mention, however, that the instructor desires to "open a trial site for teaching Chinese in the WWW," which is exactly what does *not* interest this user. He therefore takes two actions for this page: he chooses "include keyword" for the term "course," and "exclude keyword" for the term "WWW." He again requests a new search iteration.
4. The resulting hit list now contains the page "Chinese Conversation Class, Scientific Methods, and Curriculum" (at http://chinese-school.netfirms.com/curriculum.html), which is quite close to the information this user is seeking. This page advertises the Los Angeles Chinese Learning Center, which is currently engaging in some experimental teaching techniques for conversational Chinese. Described here are the curriculum, textbooks, logistics, and related opportunities. In surveying the content, the user notes and highlights several words that are indicative of the page's appropriateness. First, "curriculum" strikes him as a word reflective of the kind of instructional course he seeks – serious, planned, and thoughtful. He also selects "Mandarin," since he hadn't realized until he read this page that the Chinese language encompasses multiple dialects, and he sees now that this is the one he is interested in. Finally, the word "area" seems to convey the idea of an onsite delivery that may require travel, which this user is amenable to. He includes all three terms in the current keyword list, and requests another iteration.
5. One of the top sites on the new iteration's hit list is a "Mandarin Chinese for Professionals Short Course Program" (at http://pmp.canberra.edu.au/shortcourses/dbi_mcfpi.htm), which the user immediately recognizes as the right kind of opportunity for him. He eagerly reads about this program, and then makes a phone call to learn more about the specifics. This is the likely termination point of the search process; however, if this particular course turns out to be unsuitable, the user is free to continue the search from this very point. Since the tool has already converged to a significant degree, it is probable that many more relevant pages will arise almost immediately when the process is continued.

3.2 Discussion

Notice what happened in this brief scenario. The search tool rapidly converged on the most pertinent information by simply responding to the user's corrective feedback. And the result was as much a learning and exploration process as it was a search process. When the user began, he knew only that he wanted to learn Chinese – he was not even aware of the many alternatives involved in accomplishing that task. For instance, until he found the page describing online tools, it had perhaps not even occurred to him that there were multiple techniques available for learning a language. Only once he saw this page could he critique it and then add that detail to his mental framework. Similarly, he had not recognized the fact that his desire for fluency was not universal (other people may have more modest goals), that he needed to choose a particular dialect, and that a face-to-face instructional experience might require travel.

In short, it is normally not possible for a user to specify exhaustive search criteria at the outset. He simply does not know what questions to ask, what choices there are to make, or what kinds of information are available until he begins to explore the information space.

As magic as this may seem, in fact this tool is simply mimicking what most users already do. Very often we sit down at a search engine and scratch our heads, uncertain of what to type. We timidly try a few phrases, and based on what is returned we soon realize that either our conceptions or our terminology is wrong. The initial pages we view, irrelevant though they may be, help us clarify our thinking and realize what it is we're *not* looking for. And that is the best (and often the only) way to arrive at the gem hidden deep within the mine. The tool merely operationalizes this process. It raises it to the user's attention and facilitates its operation, encouraging him to shape the navigation path based on the semantics he perceives.

Another benefit is that the tool makes it easy to add words to the search term list directly from the page being viewed. Inexperienced users are liable to throw up their hands in frustration when a search engine cannot give them what they're looking for. Most often this stems not from inadequacies of the engine, but from the user's failure to provide the right search terms. And this in turn comes from not knowing what the best terms *are*. A user needs to be encouraged to specify exactly *what* it is he likes or doesn't like about

a particular page, so that the system can adjust and improve. Our tool makes it easy to provide such feedback directly from the critiqued page.

4. Work in progress: machine learning

Our current work is to take these notions a step further and implement an assistive machine learning component. The idea is that any page contains a multitude of words that are suggestive of its content, and by considering these as a whole we ought to be able to improve the rate of convergence.

For example, an outdoorsman interested in bass fishing might begin a Google search with the single term "bass," and be greeted by a page devoted to a famous bass guitarist. Certainly, with the existing system he could simply add the word "guitar" to the excluded list and proceed with a new iteration. But it seems likely that in addition to the word "guitar," the page would also contain a whole host of other terms that are generally reflective of music rather than fishing. It may speak at length of "rock bands," "drummers," "grooves," and "concerts," all of which would be helpful for the system to know to avoid. Our aim is to relieve the user of having to select each of these terms individually by allowing him to mark the entire *page* as "on the wrong track."

Furthermore, as the user begins to designate a whole succession of pages in this way – marking each one as either relevant or irrelevant, as appropriate – the system accrues a growing collection of evidence as to the user's underlying semantic aim. Eventually, the tool will have at its disposal a half dozen or more pages that the user has marked "relevant," and a half dozen or more marked "irrelevant." By examining the overall content of each of these pages, it should be able to get an excellent idea of precisely what kind of information the user is seeking.

Note that on the surface, this is set up exactly as a standard machine learning problem. We have two classifications – "relevant to this user" and "irrelevant to this user." Then, based on a group of labeled instances (viz., the pages the user has already judged for us), we attempt to predict the classifications for unseen pages. Each time we show the user a new page, he renders a semantic judgment, and we add the new page to either the "relevant pages" list or the "irrelevant pages" list, accordingly. The amount of training data, and presumably the classifier's accuracy, grows as the user continues to use the system. Our hypothesis is that this should lead to rapid convergence.

We use surprisingly simple methods to achieve this goal, again employing Google as the underlying search component. We create a standard vector representation of each page the user views, using the basic TF/IDF algorithm on all words not included in a publicly available stopword list[7]. Porter's rule-based stemming algorithm[5] is used to conflate related words to their root meanings. Then, as the user marks pages as relevant or irrelevant, the tool accumulates two sets of vectors, one for each group. These will form the basis for relevance predictions for as yet unseen pages.

On each successive iteration we obtain Google's best hits for the query (currently, the top ten) and form a vector representation for each of these pages. We then use an ordinary Naïve Bayes classifier on each page, trained on the two groups of pages that the user has marked. This gives us a score for each page that estimates the relative likelihood that the user would perceive that page as relevant. Finally, the page with the highest score is shown to the user to critique.

Note that although this procedure combines well-known components such as vector representations and Bayesian classifiers, the resulting composition is rather unusual. The training data can hardly be said to be randomly sampled, since we are deliberately trying to steer the search towards pages that are in the "relevant" category. Hence this is quite different from the ordinary machine learning policy, which is deliberately unbiased in selecting training data. And it is also distinct from the approach taken by active learning[8], in which the system requests labels for only those instances that will give it the most information; ie., the ones closest to the classification boundary. In our case, we are not showing the user pages that we necessarily think will help us make more accurate future decisions, but rather pages that we hope are as "unambiguous" as possible; namely, pages in the exact center of the "relevant" cluster.

The algorithm could therefore be considered greedy: it eschews the potential long-term benefits of building a more accurate classifier, instead always showing the user its current best guess as to what it thinks is most relevant. Our rationale for this is simply that the goal of the system is lead the user to the most relevant data. At each iteration, therefore, if it shows the user a page that she is interested in, it has done its job. But if it shows her a page that she is *not* interested in, it will assimilate the new counterexample and hopefully correct its "misconception." Either way, the system improves and the user is shown mostly relevant information along the way. The approach therefore seems reasonable. We intend to explore our options here, however. It may turn out that deliberately presenting instances closer to the "relevant vs. irrelevant" boundary is of some benefit early in the

search process, and if so, our current work will reveal the nature of those advantages.

Another unusual feature of our approach is its combination of automated learning and explicit user direction. The Bayesian classifier is used to predict the most relevant page out of each iteration's pool of top hits. But the pool itself is obtained from a Google search, and this is dependent only upon the search terms that the user has explicitly chosen. Thus the terms that the user specifically identifies on a page are in a sense weighed more heavily than are the other terms accompanying it, since the latter play no role in determining what Google will return for the classifier to analyze. We feel this is a good tradeoff: it appropriately honors the user's choices by absolutely mandating that they be followed, while also making use of the accompanying information to more subtly refine the results presented.

Finally, our ultimate aim is to extend these techniques beyond what is traditionally thought of as information retrieval and into the realm of information space organization. The group of pages that a user identifies as "relevant" can be used for much more than retrieving additional results. It can be made into a lasting artifact that gives valuable clues into the nature of the human's perception. We envision the user's interaction giving rise to a group of semantic *categories*, each of which contains items that the user has identified as similar in some context. After all, each search process the user engages in is a quest to satisfy some semantic concept, and hence the pages that she marks as "relevant" along the way must somehow be related. The system should be able to learn something from this collection of feature vectors and begin to make generalizations about what kinds of pages satisfy the criteria. The categories can then serve as operands for higher-level operations that allow the user to view the information space more strategically. Users can compare and contrast entire collections of objects and learn something about the correlations and tendencies among them.

Much exploratory work lay ahead of us in determining the best way to manifest these concepts to the user. But we believe that just as with the basic retrieval by reformulation paradigm, semantics emerge through user interaction. Capturing and leveraging this information is a promising avenue for turning mountains of data into fruitful knowledge.

References

1. Fails, J.A. and D.R. Olsen. *Interactive machine learning* in *Proceedings of the Eighth International Conference on Intelligent User Interfaces*. 2003. Miami, Florida.

2. Grosky, W.I., D.V. Sreenath, and F. Fotouhi, *Emergent semantics and the multimedia semantic web.* SIGMOD Record, 2002. **31**(4): p. 54-58.

3. Jansen, B.J., A. Spink, J. Bateman, and T. Saracevic, *Real life information retrieval: a study of user queries on the Web.* SIGIR Forum, 1998. **32**(1): p. 5-17.

4. Neches, R., S. Abhinkar, F. Hu, R. Eleish, I.-Y. Ko, K.-T. Yao, Q. Zhu, and P. Will, "Collaborative information space analysis tools," in *D-Lib Magazine*, October 1998.

5. Porter, M.F., *An algorithm for suffix stripping.* Program, 1980. **14**(3): p. 130-137.

6. Salton, G., A. Wong, and C.S. Yang, *A vector space model for automatic indexing.* Communications of the ACM, 1975. **18**(11): p. 613-620.

7. Savoy, J., *English language stopword list*. Institut interfacultaire d'informatique, University of Neuchatel: 2004. Available at: http://www.unine.ch/info/clef/.

8. Schohn, G. and D. Cohn. *Less is more: active learning with support vector machines*. in *Proceedings of the 17th International Conference on Machine Learning*. 2000. San Francisco, California: Morgan Kaufmann.

9. Staab, S., S. Santini, F. Nack, L. Steels, and A. Maedche, "Emergent Semantics," in *IEEE Intelligent Systems, Trends and Controversies*, Jan/Feb 2002.

10. Williams, M.D., *What makes RABBIT run?* International Journal of Human-Computer Studies, 1984. **21**(4): p. 333-352.

11. Yao, K.-T., R. Neches, I.-Y. Ko, R. Eleish, and S. Abhinkar. *Synchronous and asynchronous collaborative information space analysis tools*. in *Proceedings of the International Workshop on Collaboration and Mobile Computing*. 1999. Fukushima, Japan.