The Quest for Correct Information on the Web: Hyper Search Engines

Massimo Marchiori
Department of Pure & Applied Mathematics
University of Padova
Via Belzoni 7, 35131 Padova, Italy
max@math.unipd.it

Abstract

Finding the right information on the World Wide Web is becoming a fundamental problem, since the amount of global information that the WWW contains is growing at an incredible rate. In this paper, we present a novel method for extracting from a Web object its "hyper" informative content, in contrast with current search engines, which only deal with the "textual" informative content. This method is not only valuable per se, but it is shown to be able to considerably increase the precision of current search engines, Moreover, it integrates smoothly with existing search engine technology since it can be implemented on top of every search engine, acting as a post-processor, thus automatically transforming a search engine into its corresponding "hyper" version. We also show how, interestingly, the hyper information can be usefully employed to face the search engine persuasion problem.

1. Introduction

The World Wide Web is growing at phenomenal rates, as witnessed by every recent estimation. This explosion both of Internet hosts and of people using the Web, has made crucial the problem of managing such enormous amount of information. As market studies clearly indicate, in order to survive into this informative jungle, Web users have to resort almost exclusively to search engines (automatic catalogs of the Web) and repositories (human-maintained collections of links, usually topics-based). In turn, repositories are now themselves resorting to search engines to keep their databases up-to-date. Thus, the crucial component in information management are search engines.

As all the recent studies on the subject (see e.g., [8]) report, the performance of actual search engines is far from being fully satisfactory. While technological progress has made it possible to deal reasonably with the size of the Web in terms of number of objects classified, the big problem is just to properly classify objects in response to the user's needs: in other words, to give the user the information s/he needs.

In this paper, we argue for a solution to this problem, by means of a new measure of the informative content of a Web object, namely its "hyper information." Informally, the problem with current search engines is that they look at a Web object to evaluate, almost as a piece of text. Even with extremely sophisticated techniques like those already present in some search engine scoring mechanism, this approach suffers from an intrinsic weakness: it doesn't take into account the *Web structure* the object is part of.

The power of the Web relies on its capability of redirecting the information flow via hyperlinks, so it should appear rather natural that in order to evaluate the informative content of a Web object, the Web structure has to be carefully analyzed. Instead, so far only very limited forms of analysis of an object *in* the Web space have been used, such as "visibility" (the number of links pointing to a Web object), and the gain in obtaining a more precise information has been very small.

In this paper,instead we develop a natural notion of "hyper" information which properly takes into account the Web structure an object is part of. Informally, the "overall information" of a Web object is not composed only of its static "textual information," but also other "hyper" information has to be added, which is the measure of the potential information content of a Web object with respect to the Web space. Roughly speaking, it measures how much information one can obtain using that page with a browser, and navigating starting from it.

Besides its fundamental characteristic, and future potentials, hyper information has an immediate practical impact: indeed, it works "on top" of any textual information function, manipulating its "local" scores to obtain the more complete "global" score of a Web object. This means that it provides a smooth integration with existing search engines, since it can be used to considerably improve the performance of a search engine simply by post-processing its scores.

Moreover, we show how hyper information can deal nicely with the big problem of "search engines persuasion" (tuning pages so to cheat a search engine, in order to make it give a higher rank), one of the most alarming factors degrading of the performance of most search engines.

Finally, we present the result of an extensive testing on the hyper information, used as post-processor in combination with the major search engines, and show how the hyper information is able to *considerably* improve the quality of information provided by search engines.

2. World Wide Web

In general, we consider an *(untimed) Web structure* to be a partial function of URLs to sequences of bytes. The intuition is that for each URL we can require from the Web structure the corresponding object (an HTML page, a text file, etc.). The function has to be partial because for some URLs there is no corresponding object.

In this paper we consider as Web structure the World Wide Web structure ***WWW***. Note that in general the real ***WWW*** is a timed structure, since the URL mapping varies with time (i.e. it should be written as $\boldsymbol{WWW}_t$ for each time instant *t*). However, we will work under the hypothesis that the ***WWW*** is *locally time consistent*; i.e., that there is a non-void time interval *I* such that the probability that $\boldsymbol{WWW}_t(url)=seq$, $t >= t'>=t+I$ *implies* $\boldsymbol{WWW}_{t'}(url)=seq$ is extremely high. That is to say, if at a certain time *t* an URL points to an object *seq*, then there is a time interval in which this property stays true. Note this doesn't mean that the Web structure stays the same, since new Web objects can be added.

We will also assume that all the operations resorting on the ***WWW*** structure that we will employ in this paper can be performed with extremely high probability within time *I*: this way, we are acting on a "locally consistent" time interval, and thus it is (with extremely high probability) safe to get rid of the dynamic behavior of the World Wide Web and consider it as an untimed Web structure, as we will do.

Note that these assumptions are not restrictive, since empirical observations (cf. [2]) show that ***WWW*** is indeed locally time consistent; e.g., setting *I* = 1 day (in fact, this is one of the essential reasons why the World Wide Web works). This, in turn, also implies that the second hypothesis (algorithms working in at most *I* time) is not restrictive.

A *Web object* is a pair (*url*,*seq*), made up by an URL *url* and a sequence of bytes *seq*=***WWW***(*url*).

In the sequel, we will usually consider understood the ***WWW*** Web structure in all the situations where Web objects are considered.

As usual in the literature, when no confusion arises we will sometimes talk of a link meaning the pointed Web object (for instance, we may talk about the score of a link as meaning the score of the Web object it points to).

3. Hyper vs. Non-hyper

A great problem with search engines' scoring mechanisms is that they tend to score *text* more than *hypertext*. Let us explain this better. It is well known that the essential difference between normal texts and hypertext is the relation-flow of information which is possible via hyperlinks. Thus, when evaluating the informative content of hypertext one should keep in mind this fundamental, *dynamic* behavior of hypertext. Instead, most of search engines tend to simply forget the "hyper" part of a hypertext (the links), by simply scoring its textual components; i.e., providing the *textual information* (henceforth, we say "information" in place of the more appropriate, but longer, "measure of information").

Note that ranking in a different way words appearing in the title, or between headings, etc., as all contemporary search engines do, is not in contradiction with what was said above, since title, headings and so on are not "hyper," but are simply *attributes* of the text.

More formally, the textual information of a Web object (*url*,*seq*) considers only the textual component *seq*, not taking into account the hyper information given by *url* and by *the underlying World Wide Web structure*.

A partial exception is considering the so-called *visibility* (cf. for instance [1]) of a Web object. Roughly, the visibility of a Web object is the number of other Web objects with a link to it. Thus, visibility is in a sense a measure of the importance of a Web object in the World Wide Web context. Excite and WebCrawler were the first search engines to provide higher ranks to Web objects with high visibility; they are now followed by Lycos and Magellan.

The problem is that visibility *says nothing* about the *informative content* of a Web object. The misleading assumption is that if a Web object has high visibility, then this is a sign of importance and consideration, and so *de facto* its informative content must be more valuable than other Web objects that have fewer links pointing to them. This reasoning would be correct if all Web objects were known to users and to search engines. But this assumption is clearly false. The fact is that a Web object which is not known enough, for instance for its location, is going to have very low visibility (when it is not completely neglected by search engines), regardless of its informative content which may be by far better than that of other Web objects.

In a nutshell, visibility is likely to be a synonym of *popularity*, which is something completely different than *quality*, and thus using it to gain higher score from search engines is a rather poor choice.

3.1 The Hyper Information

As said, what is really missing in the evaluation of the score of a Web object is its hyper part, that is the dynamic information content which is provided by hyperlinks (henceforth, simply links).

We call this kind of information *hyper information*: this information should be added to the *textual information* of the Web object, giving its *(overall) information* to the World Wide Web. We indicate these three kinds of information as HYPERINFO, TEXTINFO and INFORMATION, respectively. So, for every Web object A we have that INFORMATION(A) = TEXTINFO(A) + HYPERINFO(A) (note that in general these information functions depend on a specific query, that is to say they measure the informative content of a Web object with respect to a certain query: in the sequel; we will always consider such a query to be understood).

The presence of links in a Web object clearly augments the informative content with the information contained in the pointed Web objects (although we have to establish to what extent).

Recursively, links present in the pointed Web objects further contribute, and so on. Thus, in principle, the analysis of the informative content of a Web object A should involve all the Web objects that are reachable from it via hyperlinks (i.e., "navigating" in the World Wide Web).

This is clearly unfeasible in practice, so, for practical reasons, we have to stop the analysis at a certain depth, just as in programs for chess analysis we have to stop considering the moves tree after few moves. So, one should fix a certain upper bound for the *depth* of the evaluation. The definition of depth is completely natural: given a Web object O, the (relative) depth of another Web object O' is the minimum number of links that have to be activated ("clicked") in order to reach O' from O. So, saying that the evaluation has max depth k means that we consider only the Web objects at depth less than or equal to k.

By fixing a certain depth, we thus select a suitable finite "local neighborhood" of a Web object in the World Wide Web. Now we are faced with the problem of establishing the hyper information of a Web object A w.r.t. this neighborhood (the hyper information *with depth* k). We denote this information by with HYPERINFO$_{[k]}$, and the corresponding overall information (with depth k) with INFORMATION$_{[k]}$. In most cases, k will be considered understood, and so we will omit the $[k]$ subscript.

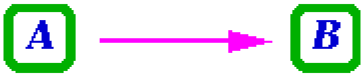
We assume that INFORMATION, TEXTINFO and HYPERINFO are functions of Web objects to nonnegative real numbers. Their intuitive meaning is that the more information a Web object has, the greater is the corresponding number.

Observe that in order to have a feasible implementation, it is necessary that all of these functions are bounded (i.e., there is a number M such that $M >=$ INFORMATION(A), for every A). So, we assume without loss of generality that TEXTINFO has an upper bound of 1, and that HYPERINFO is bounded.

3.2 Single Links

To start with, consider the simple case where we have at most one link in every Web object.

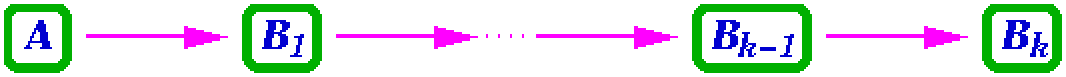
In the basic case when the depth is one, the most complicated case is when we have one link from the considered Web object A to another Web object B; i.e.,



A naïve approach is just to add the textual information of the pointed Web object. This idea, more or less, is tantamount to identifying a Web object with the Web object obtained by replacing the links with the corresponding pointed Web objects.

This approach is attractive, but not correct, since it raises a number of problems. For instance, suppose that A has almost zero textual information, while B has extremely high overall information. Using the naïve approach, A would have more informative content than B, while it is clear that the user is much more interested in B than in A.

The problem becomes even more evident when we increase the depth of the analysis: consider a situation where B_k is at depth k from A; i.e., to go from A to B one needs k "clicks", and k is large (e.g. k>10):



Let A, B_1, ..., B_{k-1} have almost zero informative content, and B_k have very high overall information. Then A would have higher overall information than B_k, which is paradoxical since A clearly has too a weak relation with B_k to be more useful than B_k itself.

The problem essentially is that the textual information pointed by a link cannot be considered as *actual*, since it is *potential*: for the user there is a *cost* for retaining the textual information pointed by a link (click and...wait).

The solution to these two factors is: the contribution to the hyper information of a Web object at depth k is not simply its textual information, but it is its textual information diminished via a fading factor depending on its depth; i.e., on "how far" is the information from the user (how many clicks s/he has to perform).

Our choice about the law regulating this fading function is that textual information fades exponentially with the depth; i.e., the contribution to the hyper information of A given by an object B at depth k is $\mathbf{F}^{k}\cdot\text{TEXTINFO}(B)$, for a suitable fading factor $\mathbf{F}$ ($0<\mathbf{F}<1$).

Thus, in the above example, the hyper information of A is not $\text{TEXTINFO}(B_1)+\ldots+\text{TEXTINFO}(B_k)$, but $\mathbf{F}\cdot\text{TEXTINFO}(B_1) + \mathbf{F}^2\cdot\text{TEXTINFO}(B_2) + \ldots + \mathbf{F}^k\cdot\text{TEXTINFO}(B_k)$, so that the overall information of A is not necessarily greater than that of B_k.

As an aside, note that when calculating the overall information of a Web object A, its textual information can be nicely seen as a special degenerate case of hyper information, since $\text{TEXTINFO}(A)=\mathbf{F}^0\cdot\text{TEXTINFO}(A)$ (viz., the object is at "zero distance" from itself).

3.3 More Motivations

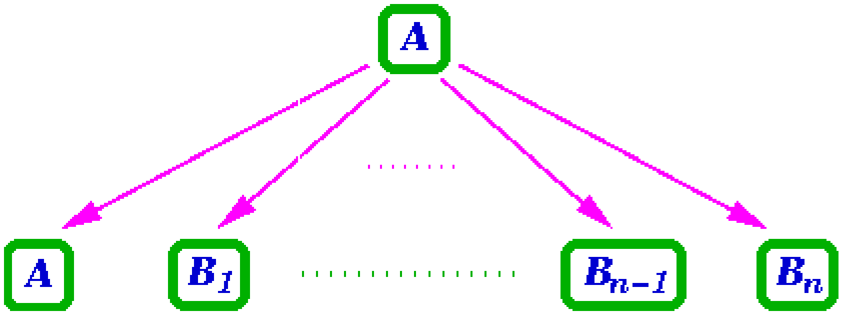
There is still one point to clarify. We have introduced an exponential fading law, and seen how it behaves well with respect to our expectations. But one could ask why the fading function has to be exponential, and not of a different form, for instance polynomial (e.g., multiplying by $\mathbf{F}/k$ instead of by $\mathbf{F}^k$). Let us consider the overall information of a Web object with depth k: if we have $\boxed{A} \longrightarrow \boxed{B}$, it is completely reasonable to assume that the overall information of A (with depth k) is given by the textual content of A plus the faded overall information of B (with depth k-1). It is also reasonable to assume that this fading can be approximated by multiplying by an appropriate fading constant $\mathbf{F}$ ($0<\mathbf{F}<1$). So, we have the recursive relation

$$\text{INFORMATION}_{[k]}(A)= \mathbf{F}\cdot\text{INFORMATION}_{[k\text{-}1]}(B)$$

Since for every object A we readily have $\text{INFORMATION}_{[0]}(A)=\text{TEXTINFO}(A)$, eliminating recursion we obtain that if $\boxed{A} \longrightarrow \boxed{B_1} \longrightarrow \cdots \longrightarrow \boxed{B_{k-1}} \longrightarrow \boxed{B_k}$ then $\text{INFORMATION}(A) = \text{TEXTINFO}(A)+\mathbf{F}\cdot(\text{TEXTINFO}(B_1)+\mathbf{F}\cdot(\text{TEXTINFO}(B_2)+\mathbf{F}\cdot(\ldots\text{TEXTINFO}(B_{\mathrm{k}})))) = \text{TEXTINFO}(A) + \mathbf{F}\cdot\text{TEXTINFO}(B_1) + \mathbf{F}^2\cdot\text{TEXTINFO}(B_2) + \ldots + \mathbf{F}^k\cdot\text{TEXTINFO}(B_k)$, which is just the exponential fading law that we have adopted.

3.4 Multiple Links

Now we turn to the case where there is more than one link in the same Web object. For simplicity, we assume the depth is one. So, suppose one has the following situation



What is the hyper information in this case? The easiest answer, just sum the contribution of every link (i.e. **F**·TEXTINFO(B_1) + ... + **F**·TEXTINFO(B_n)), isn't feasible since we want the hyper information to be *bounded*.

This would seem in contradiction with the interpretation of a link as potential information that we have given earlier: if one has many links that can be activated, then one has all of their potential information. However, this paradox is only apparent: the user cannot get all the links at the same time, but has to *sequentially select* them. In other words, *nondeterminism has a cost*. So, in the best case the user will select the most informative link, and then the second more informative one, and so on. Suppose for example that the more informative link is B_1, the second one is B_2 and so on (i.e., we have TEXTINFO(B_1) >= TEXTINFO(B_2) >= ... >= TEXTINFO(B_n)). Thus, the hyper information is **F**·TEXTINFO(B_1) (the user selects the best link) plus $\mathbf{F}^2$·TEXTINFO(B_2) (the second time, the user selects the second best link) and so on, that is to say

$$\mathbf{F}\cdot\text{TEXTINFO}(B_1) + \ldots + \mathbf{F}^n\cdot\text{TEXTINFO}(B_n)$$

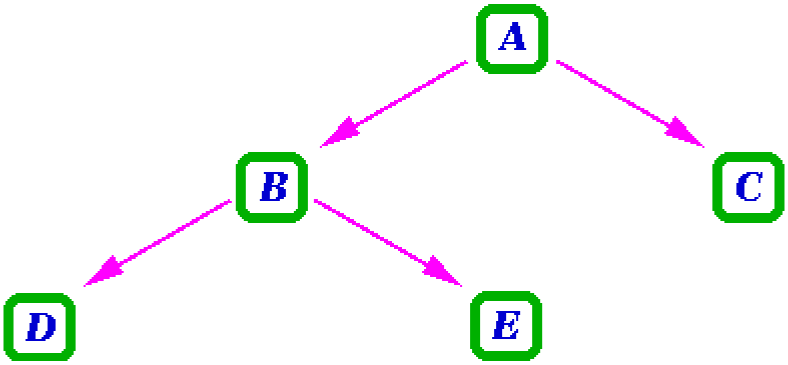
Nicely, evaluating the score this way gives a bounded function, since for any number of links, the sum cannot be greater than **F**/(**F**+1).

Note that we chose the best sequence of selections, since hyper information is the best "potential" information, so we have to assume the user does make the best choices: we cannot use, e.g., a random selection of the links, or even other functions like the average between the contributions of each link, since we cannot impose that every link has to be relevant. For instance, if we did so, accessory links with zero score (e.g. think of the "powered with Netscape"-links) would devalue by far the hyper information even in the presence of highly scored links, while those accessory links should simply be ignored (as the above method, consistently, does).

3.5 The General Case

The general case (multiple links in the same page and arbitrary depth *k*) is treated accordingly to what was previously seen. Informally, all the Web objects at depth less than or equal to *k* are ordered with a "sequence of selections" such that the corresponding hyper information is the highest one.

For example, consider the following situation:



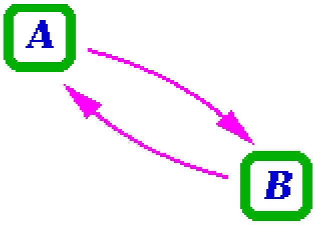
with **F**=0.5, TEXTINFO(*B*)=0.4, TEXTINFO(*C*)=0.3, TEXTINFO(*D*)=0.2, TEXTINFO(*E*)=0.6.

Then via a "sequence of selections" we can go from *A* to *B*, to *C*, to *E* and then to *D*, and this is readily the best sequence that maximizes the hyper information, which is 0.5·TEXTINFO(*B*) + 0.5^2·TEXTINFO(*C*) + 0.5^3·TEXTINFO(*E*) + 0.5^4·TEXTINFO(*D*) (=0.3625).

4. Refining the Model

There are a number of subtleties that for clarity of exposition we haven't considered so far.

The first big problem comes from possible duplications. For instance, consider the case of *backward* links: suppose that a user has set up two pages *A* and *B*, with a link in from *A* to *B*, and then a "come back" link in *B* from *B* to *A*:



This means that we have a kind of recursion, and that the textual information of *A* is repeatedly added (although with increasingly higher fading) when calculating the hyper information of *A* (this is because from *A* we can navigate to *B* and then to *A* and so on...). Another example is given by *duplicating links* (e.g., two links in a Web object pointing to the same Web object). The solution to avoiding these kinds of problems, is not to consider all the "sequence of selections", but only those *without repetitions*. This is consistent with the intuition of hyper information to measure in a sense how far a Web object is from a page: if one has already reached an object, s/he already has got its information, and so it makes no sense to get it again.

Another important issue is that the same definition of link in a Web object is fairly from trivial. A link present in a Web object *O* is said to be *active* if the Web objects it points to can be accessed by viewing (*url*,*seq*) with an HTML browser (e.g., Netscape Navigator or Microsoft Internet Explorer). This means, informally, that once we view *O* with the browser, we can activate the link by clicking over it. The previous definition is rather operational, but it is much more intuitive than a formal technical definition which can be given by tediously specifying all the possible cases according to the HTML specification (note a problem complicating a formal analysis is that one cannot assume that *seq* is composed by legal HTML code, since browsers are error-tolerating).

Thus, the links mentioned in the paper should be only the active ones.

Another subtlety is given by duplicating links (e.g. two links in a Web object pointing to the same Web object). In these cases, checks are necessary in order to avoid considering the same information more than once.

Also, there are many different kinds of links, and each of them would require a specific treatment. For instance:

- *Local* links (links pointing to some point in the same Web object, using the #-specifier) should be readily ignored.
- *Frame* links should be automatically expanded; i.e., if *A* has a frame link to *B*, then this link should be replaced with a proper expansion of *B* inside *A* (since a frame link is automatically activated, its pointed Web object is just part of the original Web object, and the user does not see any link at all).
- Other links that are automatically activated are for instance the *image* links; i.e., source links of image tags, the *background* links and so on; they should be treated analogously to frame links. However, since it is usually a very hard problem to recover some useful information from images (cf. [6]), in a practical implementation of the hyper information all these links can be ignored (note this doesn't mean that the textual information function has to ignore such links, since it can, e.g., take into account the name of the pictures). This also happens for active links pointing to images, movies, sounds etc., which can be syntactically identified by their file extensions .gif, .jpg, .tiff, .avi, .wav and so on.

4.1 Search Engines Persuasion

A big problem that search engines have to face is the phenomenon of so-called *sep* (*s*earch *e*ngines *p*ersuasion). Indeed, search engines have become so important in the advertising market that it has become essential for companies to have their pages listed in top positions of search engines, in order to get a significant Web-based promotion. Starting with the already pioneering work of Rhodes ([5]), this phenomenon is now boosting at such a rate as to have provoked serious problems for search engines, and has revolutioned the Web design companies, which are now specifically asked not only to design good Web sites, but also to make them rank high in search engines. A vast number of new companies was born just to make customer Web pages as visible as possible. More and more companies, like Exploit, Allwilk, Northern Webs, Ryley & Associates, etc., explicitly study ways to rank a page highly in search engines. OpenText arrived even to sell "preferred listings;" i.e., assuring that a particular entry will stay in the top ten for some time, a policy that has provoked some controversies (cf. [9]).

This has led to a bad performance degradation of search engines, since an increasingly high number of pages is designed to have an artificially high textual content. The phenomenon is so serious that search engines like InfoSeek and Lycos have introduced penalties to face the most common of these persuasion techniques, "spamdexing" ([4, 7, 8]); i.e, the artificial repetition of relevant keywords. Despite these efforts, the situation is getting worse and worse, since more sophisticated techniques have been developed, which analyze the behavior of search engines and tune the pages accordingly. This has also led to the situation where search engine maintainers tend to assign penalties to pages that rank "too high", and at the same time provide less and less details on their information functions just in order to prevent this kind of tuning, thus in many cases penalizing pages of high informative value that were not designed with "persuasion" intentions. For a comprehensive study of the sep phenomenon, we refer to [3].

The hyper methodology that we have developed is able to a certain extent to cope nicely with the sep problem. Indeed, maintainers can keep details of their TEXTINFO function hidden, and make public the information that they are using a HYPERINFO function.

The only precaution is to distinguish between two fundamental types of links, assigning different fading factors to each of them. Suppose wew have a Web object (*url*,*seq*). A link contained in *seq* is called *outer* if it does not have the same domain of *url*, and *inner* in the other case. That is to say, inner links of a Web objects point to Web objects in the same site (its "local world", so to speak), while outer links point to Web objects of other sites (the "outer world").

Now, inner links are dangerous from the sep point of view, since they are in direct control of the site maintainer. For instance, a user that wants to artificially increase the hyper information of a Web object A could set up a very similar Web object B (i.e. such that TEXTINFO(A)~TEXTINFO(B)), and put a link from A to B: this would increase the score of A by roughly $\mathbf{F}\cdot$TEXTINFO(A).

On the other hand, outer links do not present this problem since they are out of direct control and manipulation (at least in principle, cf. [1], at present the inter-connectivity is rather poor, since almost 80% of sites contain no outer link (!), and a relatively small number of Web sites is carrying most of the load of hypertext navigation.

Using hyper information thus forces the sites that want to rank better to *improve their connectivity*, improving the overall Web structure.

5. Testing

The hyper information has been implemented as *post- processor* of the main search engines now available; i.e., remotely querying them, extracting the corresponding scores (i.e., their TEXTINFO function), and calculating the hyper information and therefore the overall information.

Indeed, as stated in the introduction, one of the most appealing features of the hyper methodology is that it can be implemented "on top" of existing scoring functions. Note that, strictly speaking, scoring functions employing visibility, like those of Excite, WebCrawler and Lycos, are not pure "textual information." However, this is of little importance; although visibility, as shown, is not a good choice, it provides information which is disjoint to the hyper information, and so we can view such scoring functions as providing purely textual information slightly perturbed (improved?) with some other kind of ***WWW***-based information.

The search engines for which a post-processor was developed were: Excite, HotBot, Lycos, WebCrawler, and OpenText. This list includes all of the major search engines, but for AltaVista and InfoSeek, which unfortunately do not give the user access to the scores, and thus cannot be remotely post-processed. The implemented model included all the accessory refinements seen in the section "Refining the Model".

We then conducted some tests in order to see how the values of $\mathbf{F}_{in}$ and $\mathbf{F}_{out}$ affected the quality of the hyper information. The depth and fading parameters can be flexibly employed to tune the hyper information. However, it is important for its practical use that the behavior of the hyper information is tested when the depth and the fading factors are fixed in advance.

We randomly selected 25 queries, collected all the corresponding rankings from the aforementioned search engines, and then ran several tests in order to maximize the effectiveness of the hyper information. We finally selected the following global parameters for the hyper information: $\mathbf{F}_{in}$=0, $\mathbf{F}_{out}$=0.75, and depth one. Although the best values were slightly different ($\mathbf{F}_{in}$=0.1, depth two), the differences were almost insignificant. On the other hand, choosing $\mathbf{F}_{in}$=0 and depth one had great advantages in terms of execution speed of the hyper information.

After this setting, we chose 25 other queries, and tested again the bounty of the hyper information with these fixed parameters; the results showed that these initial settings also gave extremely good results for these new set of queries.

While our personal tests clearly indicated a considerable increase of precision for the search engines, there was obviously the need to get external confirmation of the effectiveness of the approach. Also, although the tests indicated that slightly perturbing the fading factors did not substantially affect the hyper information, the bounty of the values of the fading factors could somehow have been dependent on our way of choosing queries.

Therefore, we performed a deeper and more challenging test. We considered a group of thirty persons, and asked each of them to arbitrarily select five different topics to search in the World Wide Web.

Then each person had to perform the following test: s/he had to search for relevant information on each chosen topic by inputting suitable queries to our post-processor, and give an evaluation mark to the obtained ranking lists. The evaluation consisted in an integer ranging from 0 (terrible ranking, of no use), to 100 (perfect ranking).

The post-processor was implemented as a cgi-script: the prompt asks for a query, and returns two ranking lists relative to that query, one per column in the same page (frames are used). In one column there is the original top ten of the search engine. In the other column, the top ten of the ranking obtained by: 1) taking the first 100 items of the ranking of the search engine, and 2) post-processing them with the hyper information.

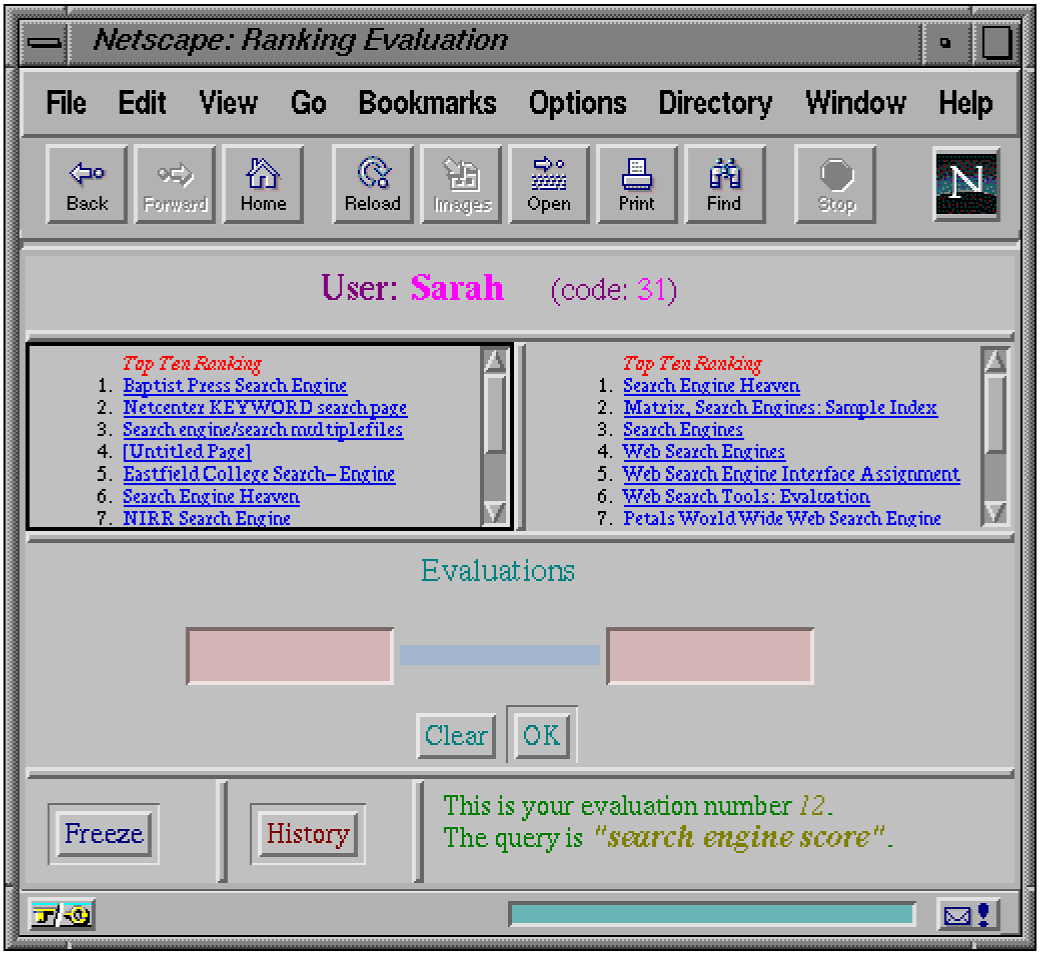
We made serious efforts to provide each user with the most natural conditions in order to evaluate the rankings. Indeed, to avoid "psychological pressure" we provided each user with a password, so that each user could remotely perform the test on his/her favorite terminal, in every time period (the test process could be frozen via a special button, terminating the session, and resumed whenever desired), with all the time necessary to evaluate a ranking (e.g., possibly via looking at each link using the browser). Besides the evident beneficial effects on the bounty of the data, this also had the side-effect that we hadn't to take care of each test personally, something which could have been extremely time consuming.

Another *extremely important* point is that the post-processor was implemented to make the test *completely blind*. This was achieved in the following way:

Each time a query was run, the order in which the five search engines had to be applied was randomly chosen. Then, for each of the five search engines the following process occurred:

- The data obtained by the search engine were "filtered" and presented in a standard "neutral" format. The same format was adopted for the post-processed data, so that the user couldn't see any difference between the two rankings except for their links content.
- The columns in which view the original search engine results and the post-processed results were randomly chosen.
- The users were provided with only the information that we were performing a test on evaluating several different scoring mechanisms, and that the choice of the column where to view each ranking was random. This also avoided possible "chain" effects, where a user, after having given higher scores to a fixed column for a certain number of times, can be unconsciously led to give it bonuses even if it actually doesn't deserve them.

Figure 1 shows a sample snapshot of a session, where the selected query is "search engine score"; the randomly selected search engine in this case was HotBot, and its original ranking (filtered) is in the left column.



User: Sarah (code: 31)

Top Ten Ranking

1. Baptist Press Search Engine
2. Netcenter KEYWORD search page
3. Search engine/search multiplefiles
4. [Untitled Page]
5. Eastfield College Search–Engine
6. Search Engine Heaven
7. NIRR Search Engine

Top Ten Ranking

1. Search Engine Heaven
2. Matrix, Search Engines: Sample Index
3. Search Engines
4. Web Search Engines
5. Web Search Engine Interface Assignment
6. Web Search Tools: Evaluation
7. Petals World Wide Web Search Engine

Evaluations

This is your evaluation number *12*.
The query is *"search engine score"*.

Figure 1. Snapshot from a test session.

The final results of the test are pictured in Figure 2. As it can be seen, the chart clearly shows (especially in view of the blind evaluation process), that the hyper information is able to considerably improve the evaluation of the informative content.

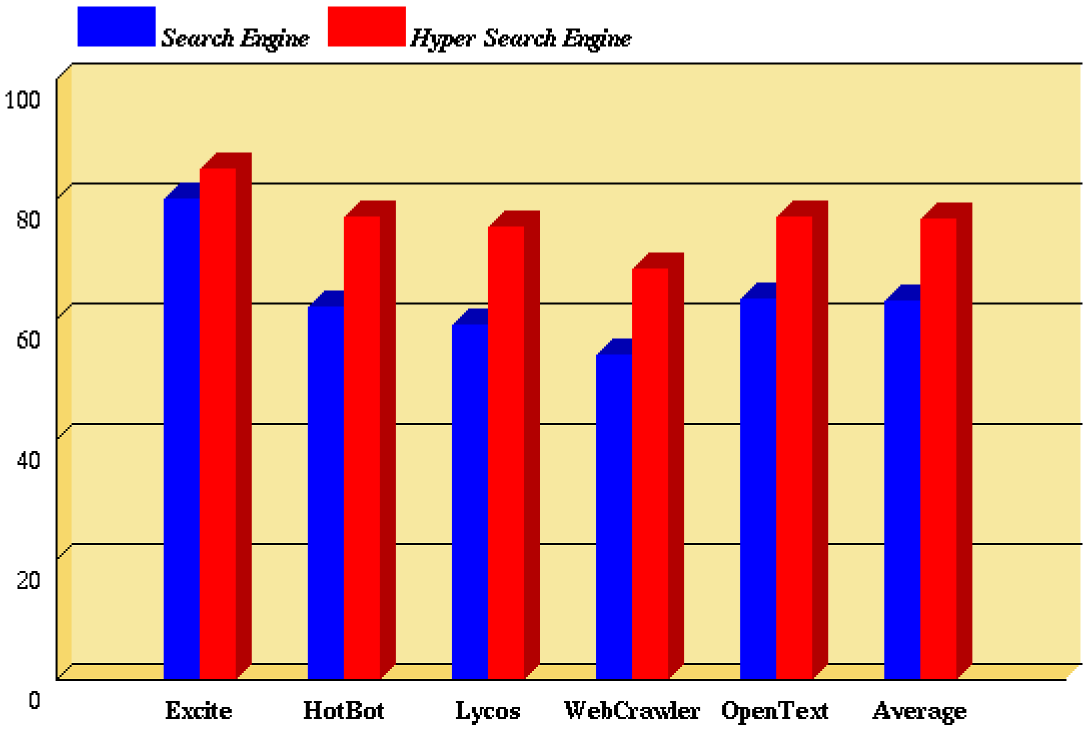


Figure 2: evaluation of search engines vs. their hyper versions.

The precise data regarding the evaluation test are shown in Table 1:

Table 1.

	Excite	HotBot	Lycos	WebCrawler	OpenText	Average
Normal	80.1	62.2	59.0	54.2	63.4	63.2
Hyper	85.2	77.3	75.4	68.5	77.1	76.7

The next table shows the evaluation increment for each search engine w.r.t. its hyper version, and the corresponding standard deviation:

Table 2.

	Excite	HotBot	Lycos	WebCrawler	OpenText	Average
Evaluation increment	+5.1	+15.1	+16.4	+14.3	+13.7	+12.9
Standard deviation	2.2	4.1	3.6	1.6	3.0	2.9

As it can be seen, the small standard deviations are further empirical evidence of the superiority of the hyper search engines over their non-hyper versions.

6. Conclusion

We have presented a new way to considerably improve the quality of search engines evaluations by considering the "hyper information" of Web objects. We have shown that besides its practical success, the method presents several other interesting features:

- It permits focusing separately on the "textual" and "hyper" components.

- It permits improvent of current search engines in a smooth way, since it works "on top" of any existing scoring function.
- It can be used for locally improving performances of search engines, just using a local post-processor on the client side (as done in this paper)
- Analogously, it can be used to improve so-called "meta searchers," such as SavySearch, MetaCrawler, WebCompass, NlightN and so on. These suffer even more from the problem of good scoring, since they have to mix data coming from different search-engines, and the result is in most of cases not very readable. The method can considerably improve effectiveness and utility for the user of such meta-engines.
- Last but not least, it behaves extremely well with respect to search engines persuasion attacks.

References

[1]
Tim Bray. Measuring the Web. *Fifth International World Wide Web Conference*, May, Paris, 1996.

[2]
Massimo Marchiori. The Hyper Information: Theory and Practice. Technical Report, Dept. of Pure & Applied Mathematics, University of Padova, 1996, to appear.

[3]
Massimo Marchiori. Security of World Wide Web Search Engines. *Third International Conference on Reliability, Quality and Safety of Software-Intensive Systems*. Athens, Greece, Chapman & Hall, 1997.

[4]
Kathleen Murphy. Cheaters Never Win. *Web Week*, May, 1996.

[5]
Jim Rhodes. The Art of Business Web Site Promotion, deadlock Design, 1996.

[6]
Stan Sclaroff. World Wide Web Image Search Engines. *NSF Workshop on Visual Information Management*, Cambridge, Massachusetts, 1995.

[7]
Danny Sullivan. The Webmaster's Guide to Search Engines and Directories. *Calafia Consulting*, 1996.

[8]
Gus Venditto. Search Engine Showdown. *Internet World*, volume 7, number 5, May, 1996.

[9]
Nick Wingfield. Engine sells results, draws fire. *C|net Inc.*, June, 1996.

Hyper References

Allwilk:
http://www.allwilk.com
Altavista:
http://www.altavista.com
Background:
http://www.w3.org/pub/WWW/TR/REC-html32.html#body
Calafia Consulting:
http://calafia.com

Chapman & Hall:

http://www.thomson.com:8866/chaphall/default.html

Cheaters Never Win:

http://www.Webweek.com/96May20/undercon/cheaters.html

C|net Inc.:

http://www.news.com

Deadlock Design:

http://deadlock.com/~deadlock

Domain:

http://www.dns.net/dnsrd

Engine sells results, draws fire:

http://www.news.com/News/Item/0,4,1635,00.html

Excite:

http://www.excite.com

Exploit:

http://www.exploit.com

Fifth International World Wide Web Conference:

http://www.w3.org/pub/Conferences/WWW5/

Frames:

http://home.netscap e.com/assist/net_sites/frames.html

Heading:

http://www.w3.org/pub/WWW/TR/REC-html32.html#headings

HotBot:

http://www.hotbot.com

HTML:

http://www.w3.org/pub/WWW/MarkUp/

Image:

http://www.w3.org/pub/WWW/TR/REC-html32.html#img

InfoSeek:

http://www.infoseek.com

Internet Explorer:

http://www.microsoft.com/ie

Internet World:

http://www.internetworld.com

Internet World, volume 7, number 5:

http://www.internetworld.com/1996/05/toc.html

Lycos:

http://www.lycos.com

Magellan:

http://www.mckinley.com

Measuring the Web:

http://www5conf.inria.fr/fich_html/papers/P9/Overview.html

MetaCrawler:

http://www.metacrawler.com

Microsoft:

http://www.microsoft.com

Navigator:

http://www.netscape.com/comprod/products/navigator/version_3.0/index.html

Netscape:

http://www.netscape.com

NlightN:

http://www.nlightn.com

Northern Webs:

http://www.digital-cafe.com/~Webmaster/norWeb01.htm

OpenText:
http://www.opentext.com

Powered with Netscape:
http://www.netscape.com/comprod/products/navigator/version_3.0/images/netnow3.gif

Ryley & Associates:
http://www.ryley.com

SavySearch:
http://www.cs.colostate.edu/~dreiling/smartform.html

Search Engine Showdown:
http://www.internetworld.com/1996/05/showdown.html

The Art of Business Web Sites Promotion:
http://deadlock.com/~deadlock/promote

The Webmaster's Guide to Search Engines and Directories:
http://calafia.com/Webmasters/

Title:
http://www.w3.org/pub/WWW/TR/REC-html32.html#title

URL:
http://www.w3.org/pub/WWW/Addressing/rfc1738.txt

WebCompass:
http://arachnid.qdeck.com/qdeck/demosoft/Webcompass_lite/

WebCrawler:
http://www.Webcrawler.com

Web Week:
http://www.Webweek.com

World Wide Web Image Search Engines:
http://www.cs.bu.edu/techreports/95-016-www-image-search- engines.ps.Z