The PageRank Citation Ranking: Bringing Order to the Web

January 29, 1998

Abstract

The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes PageRank, a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them.

We compare PageRank to an idealized random Web surfer. We show how to efficiently compute PageRank for large numbers of pages. And, we show how to apply PageRank to search and to user navigation.

1 Introduction and Motivation

The World Wide Web creates many new challenges for information retrieval. It is very large and heterogeneous. Current estimates are that there are over 150 million web pages with a doubling life of less than one year. More importantly, the web pages are extremely diverse, ranging from "What is Joe having for lunch today?" to journals about information retrieval. In addition to these major challenges, search engines on the Web must also contend with inexperienced users and pages engineered to manipulate search engine ranking functions.

However, unlike "flat" document collections, the World Wide Web is hypertext and provides considerable auxiliary information on top of the text of the web pages, such as link structure and link text. In this paper, we take advantage of the link structure of the Web to produce a global "importance" ranking of every web page. This ranking, called PageRank, helps search engines and users quickly make sense of the vast heterogeneity of the World Wide Web.

1.1 Diversity of Web Pages

Although there is already a large literature on academic citation analysis, there are a number of significant differences between web pages and academic publications. Unlike academic papers which are scrupulously reviewed, web pages proliferate free of quality control or publishing costs. With a simple program, huge numbers of pages can be created easily, artificially inflating citation counts. Because the Web environment contains competing profit seeking ventures, attention getting strategies evolve in response to search engine algorithms. For this reason, any evaluation strategy which counts replicable features of web pages is prone to manipulation. Further, academic papers are well defined units of work, roughly similar in quality and number of citations, as well as in their purpose – to extend the body of knowledge. Web pages vary on a much wider scale than academic papers in quality, usage, citations, and length. A random archived message posting asking an obscure question about an IBM computer is very different from the IBM home page. A research article about the effects of cellular phone use on driver attention is very different from an advertisement for a particular cellular provider. The average web page quality experienced by a user is higher than the quality of the average web page. This is because the simplicity of creating and publishing web pages results in a large fraction of low quality web pages that users are unlikely to read.

There are many axes along which web pages may be differentiated. In this paper, we deal primarily with one - an approximation of the overall relative importance of web pages.

1.2 PageRank

In order to measure the relative importance of web pages, we propose PageRank, a method for computing a ranking for every web page based on the graph of the web. PageRank has applications in search, browsing, and traffic estimation.

Section 2 gives a mathematical description of PageRank and provides some intuitive justification. In Section 3, we show how we efficiently compute PageRank for as many as 518 million hyperlinks. To test the utility of PageRank for search, we built a web search engine called Google (Section 5). We also demonstrate how PageRank can be used as a browsing aid in Section 7.3.

2 A Ranking for Every Page on the Web

2.1 Related Work

There has been a great deal of work on academic citation analysis [Gar95]. Goffman [Gof71] has published an interesting theory of how information flow in a scientific community is an epidemic process.

There has been a fair amount of recent activity on how to exploit the link structure of large hypertext systems such as the web. Pitkow recently completed his Ph.D. thesis on "Characterizing World Wide Web Ecologies" [Pit97, PPR96] with a wide variety of link based analysis. Weiss discuss clustering methods that take the link structure into account [WVS$^+$96]. Spertus [Spe97] discusses information that can be obtained from the link structure for a variety of applications. Good visualization demands added structure on the hypertext and is discussed in [MFH95, MF95]. Recently, Kleinberg [Kle98] has developed an interesting model of the web as Hubs and Authorities, based on an eigenvector calculation on the co-citation matrix of the web.

Finally, there has been some interest in what "quality" means on the net from a library community [Til].

It is obvious to try to apply standard citation analysis techniques to the web's hypertextual citation structure. One can simply think of every link as being like an academic citation. So, a major page like http://www.yahoo.com/ will have tens of thousands of backlinks (or citations) pointing to it.

This fact that the Yahoo home page has so many backlinks generally imply that it is quite important. Indeed, many of the web search engines have used backlink count as a way to try to bias their databases in favor of higher quality or more important pages. However, simple backlink counts have a number of problems on the web. Some of these problems have to do with characteristics of the web which are not present in normal academic citation databases.

2.2 Link Structure of the Web

While estimates vary, the current graph of the crawlable Web has roughly 150 million nodes (pages) and 1.7 billion edges (links). Every page has some number of forward links (outedges) and backlinks (inedges) (see Figure 1). We can never know whether we have found all the backlinks of a particular page but if we have downloaded it, we know all of its forward links at that time.

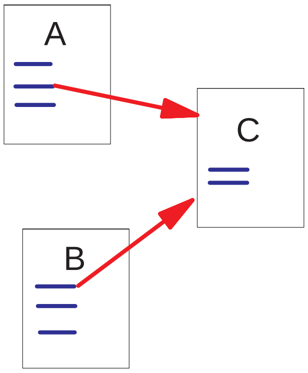

Figure 1: A and B are Backlinks of C

Web pages vary greatly in terms of the number of backlinks they have. For example, the Netscape home page has 62,804 backlinks in our current database compared to most pages which have just a few backlinks. Generally, highly linked pages are more "important" than pages with few links. Simple citation counting has been used to speculate on the future winners of the Nobel Prize [San95]. PageRank provides a more sophisticated method for doing citation counting.

The reason that PageRank is interesting is that there are many cases where simple citation counting does not correspond to our common sense notion of importance. For example, if a web page has a link off the Yahoo home page, it may be just one link but it is a very important one. This page should be ranked higher than many pages with more links but from obscure places. PageRank is an attempt to see how good an approximation to "importance" can be obtained just from the link structure.

2.3 Propagation of Ranking Through Links

Based on the discussion above, we give the following intuitive description of PageRank: a page has high rank if the sum of the ranks of its backlinks is high. This covers both the case when a page has many backlinks and when a page has a few highly ranked backlinks.

2.4 Definition of PageRank

Let u be a web page. Then let F_u be the set of pages u points to and B_u be the set of pages that point to u. Let $N_u = |F_u|$ be the number of links from u and let c be a factor used for normalization (so that the total rank of all web pages is constant).

We begin by defining a simple ranking, R which is a slightly simplified version of PageRank:

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

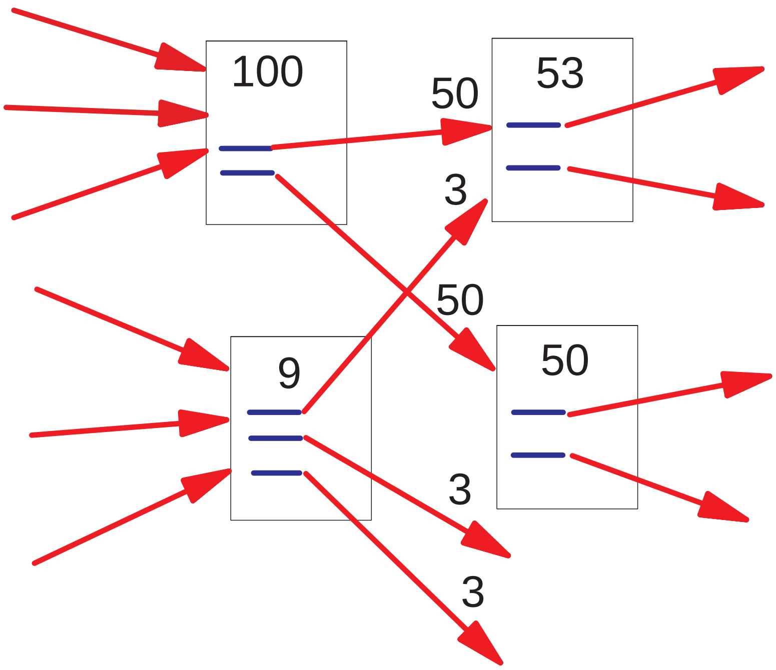

Figure 2: Simplified PageRank Calculation

This formalizes the intuition in the previous section. Note that the rank of a page is divided among its forward links evenly to contribute to the ranks of the pages they point to. Note that $c < 1$ because there are a number of pages with no forward links and their weight is lost from the system (see section 2.7). The equation is recursive but it may be computed by starting with any set of ranks and iterating the computation until it converges. Figure 2 demonstrates the propagation of rank from one pair of pages to another. Figure 3 shows a consistent steady state solution for a set of pages.

Stated another way, let A be a square matrix with the rows and column corresponding to web pages. Let $A_{u,v} = 1/N_u$ if there is an edge from u to v and $A_{u,v} = 0$ if not. If we treat R as a vector over web pages, then we have $R = cAR$. So R is an eigenvector of A with eigenvalue c. In fact, we want the dominant eigenvector of A. It may be computed by repeatedly applying A to any nondegenerate start vector.

There is a small problem with this simplified ranking function. Consider two web pages that point to each other but to no other page. And suppose there is some web page which points to one of them. Then, during iteration, this loop will accumulate rank but never distribute any rank (since there are no outedges). The loop forms a sort of trap which we call a rank sink.

To overcome this problem of rank sinks, we introduce a rank source:

Definition 1 *Let $E(u)$ be some vector over the Web pages that corresponds to a source of rank. Then, the PageRank of a set of Web pages is an assignment, R', to the Web pages which satisfies*

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u) \tag{1}$$

such that c is maximized and $||R'||_1 = 1$ ($||R'||_1$ denotes the L_1 norm of R').

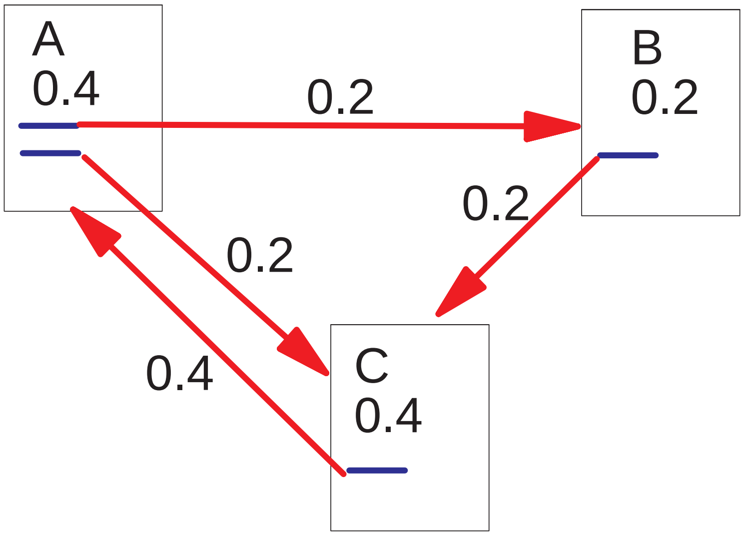

Figure 3: Simplified PageRank Calculation

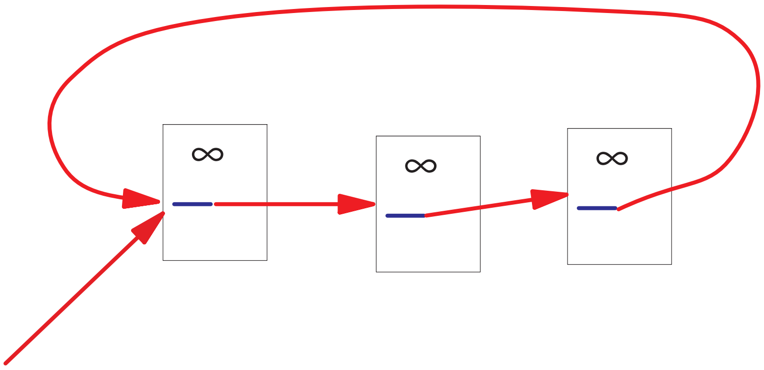

Figure 4: Loop Which Acts as a Rank Sink

where $E(u)$ is some vector over the web pages that corresponds to a source of rank (see Section 6). Note that if E is all positive, c must be reduced to balance the equation. Therefore, this technique corresponds to a decay factor. In matrix notation we have $R' = c(AR' + E)$. Since $||R'||_1 = 1$, we can rewrite this as $R' = c(A + E \times \mathbf{1})R'$ where $\mathbf{1}$ is the vector consisting of all ones. So, R' is an eigenvector of $(A + E \times \mathbf{1})$.

2.5 Random Surfer Model

The definition of PageRank above has another intuitive basis in random walks on graphs. The simplified version corresponds to the standing probability distribution of a random walk on the graph of the Web. Intuitively, this can be thought of as modeling the behavior of a "random surfer". The "random surfer" simply keeps clicking on successive links at random. However, if a real Web surfer ever gets into a small loop of web pages, it is unlikely that the surfer will continue in the loop forever. Instead, the surfer will jump to some other page. The additional factor E can be viewed as a way of modeling this behavior: the surfer periodically "gets bored" and jumps to a random page chosen based on the distribution in E.

So far we have left E as a user defined parameter. In most tests we let E be uniform over all web pages with value α. However, in Section 6 we show how different values of E can generate "customized" page ranks.

2.6 Computing PageRank

The computation of PageRank is fairly straightforward if we ignore the issues of scale. Let S be almost any vector over Web pages (for example E). Then PageRank may be computed as follows:

$$
\begin{aligned}
R_0 &\leftarrow S \\
\text{loop} :& \\
R_{i+1} &\leftarrow AR_i \\
d &\leftarrow ||R_i||_1 - ||R_{i+1}||_1 \\
R_{i+1} &\leftarrow R_{i+1} + dE \\
\delta &\leftarrow ||R_{i+1} - R_i||_1 \\
\text{while } \delta > \epsilon &
\end{aligned}
$$

Note that the d factor increases the rate of convergence and maintains $||R||_1$. An alternative normalization is to multiply R by the appropriate factor. The use of d may have a small impact on the influence of E.

2.7 Dangling Links

One issue with this model is dangling links. Dangling links are simply links that point to any page with no outgoing links. They affect the model because it is not clear where their weight should be distributed, and there are a large number of them. Often these dangling links are simply pages that we have not downloaded yet, since it is hard to sample the entire web (in our 24 million pages currently downloaded, we have 51 million URLs not downloaded yet, and hence dangling). Because dangling links do not affect the ranking of any other page directly, we simply remove them from the system until all the PageRanks are calculated. After all the PageRanks are calculated, they can be added back in, without affecting things significantly. Notice the normalization of the other links on the same page as a link which was removed will change slightly, but this should not have a large effect.

3 Implementation

As part of the Stanford WebBase project [PB98], we have built a complete crawling and indexing system with a current repository of 24 million web pages. Any web crawler needs to keep a database of URLs so it can discover all the URLs on the web. To implement PageRank, the web crawler simply needs to build an index of links as it crawls. While a simple task, it is non-trivial because of the huge volumes involved. For example, to index our current 24 million page database in about five days, we need to process about 50 web pages per second. Since there about about 11 links on an average page (depending on what you count as a link) we need to process 550 links per second. Also, our database of 24 million pages references over 75 million unique URLs which each link must be compared against.

Much time has been spent making the system resilient in the face of many deeply and intricately flawed web artifacts. There exist infinitely large sites, pages, and even URLs. A large fraction of web pages have incorrect HTML, making parser design difficult. Messy heuristics are used to help the crawling process. For example, we do not crawl URLs with /cgi-bin/ in them. Of course it is impossible to get a correct sample of the "entire web" since it is always changing. Sites are sometimes down, and some people decide to not allow their sites to be indexed. Despite all this, we believe we have a reasonable representation of the actual link structure of publicly accessible web.

3.1 PageRank Implementation

We convert each URL into a unique integer, and store each hyperlink in a database using the integer IDs to identify pages. Details of our implementation are in [PB98]. In general, we have implemented PageRank in the following manner. First we sort the link structure by Parent ID. Then dangling links are removed from the link database for reasons discussed above (a few iterations removes the vast majority of the dangling links). We need to make an initial assignment of the ranks. This assignment can be made by one of several strategies. If it is going to iterate until convergence, in general the initial values will not affect final values, just the rate of convergence. But we can speed up convergence by choosing a good initial assignment. We believe that careful choice of the initial assignment and a small finite number of iterations may result in excellent or improved performance.

Memory is allocated for the weights for every page. Since we use single precision floating point values at 4 bytes each, this amounts to 300 megabytes for our 75 million URLs. If insufficient RAM is available to hold all the weights, multiple passes can be made (our implementation uses half as much memory and two passes). The weights from the current time step are kept in memory, and the previous weights are accessed linearly on disk. Also, all the access to the link database, A, is linear because it is sorted. Therefore, A can be kept on disk as well. Although these data structures are very large, linear disk access allows each iteration to be completed in about 6 minutes on a typical workstation. After the weights have converged, we add the dangling links back in and recompute the rankings. Note after adding the dangling links back in, we need to iterate as many times as was required to remove the dangling links. Otherwise, some of the dangling links will have a zero weight. This whole process takes about five hours in the current implementation. With less strict convergence criteria, and more optimization, the calculation could be much faster. Or, more efficient techniques for estimating eigenvectors could be used to improve performance. However, it should be noted that the cost required to compute the PageRank is insignificant compared to the cost required to build a full text index.

4 Convergence Properties

As can be seen from the graph in Figure 4 PageRank on a large 322 million link database converges to a reasonable tolerance in roughly 52 iterations. The convergence on half the data takes roughly 45 iterations. This graph suggests that PageRank will scale very well even for extremely large collections as the scaling factor is roughly linear in $\log n$.

One of the interesting ramifications of the fact that the PageRank calculation converges rapidly is that the web is an expander-like graph. To understand this better, we give a brief overview of the theory of random walks on graphs; refer to Motwani-Raghavan [MR95] for further details. A random walk on a graph is a stochastic process where at any given time step we are at a particular node of the graph and choose an outedge uniformly at random to determine the node to visit at the next time step. A graph is said to be an expander if it is the case that every (not too large) subset of nodes S has a neighborhood (set of vertices accessible via outedges emanating from nodes in S) that is larger than some factor α times $|S|$; here, α is called the expansion factor. It is the case that a graph has a good expansion factor if and only if the largest eigenvalue is sufficiently larger than the second-largest eigenvalue. A random walk on a graph is said to be rapidly-mixing if it quickly (time logarithmic in the size of the graph) converges to a limiting distribution on the set of nodes in the graph. It is also the case that a random walk is rapidly-mixing on a graph if and only if the graph is an expander or has an eigenvalue separation.

To relate all this to the PageRank computation, note that it is essentially the determination of the limiting distribution of a random walk on the Web graph. The importance ranking of a node is essentially the limiting probability that the random walk will be at that node after a sufficiently large time. The fact that the PageRank computation terminates in logarithmic time is equivalent to saying that the random walk is rapidly mixing or that the underlying graph has a good expansion factor. Expander graphs have many desirable properties that we may be able to exploit in the future in computations involving the Web graph.

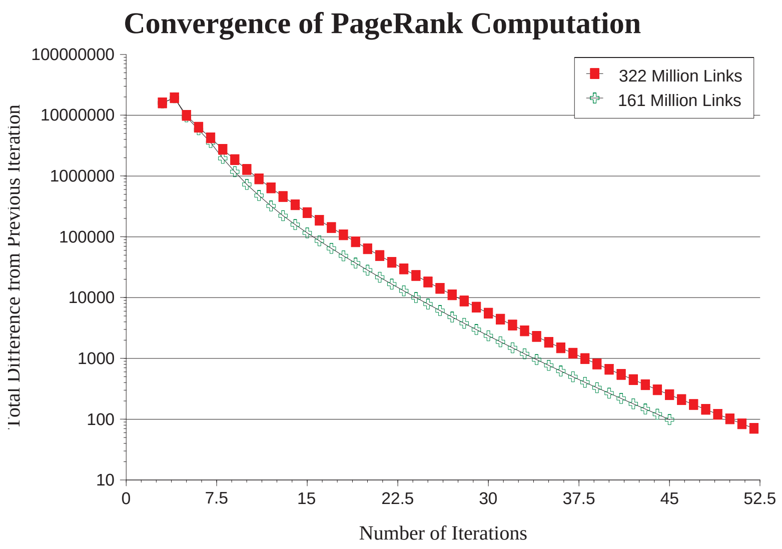

Figure 5: Rates of Convergence for Full Size and Half Size Link Databases

5 Searching with PageRank

A major application of PageRank is searching. We have implemented two search engines which use PageRank. The first one we will discuss is a simple title-based search engine. The second search engine is a full text search engine called Google [BP]. Google utilizes a number of factors to rank search results including standard IR measures, proximity, anchor text (text of links pointing to web pages), and PageRank. While a comprehensive user study of the benefits of PageRank is beyond the scope of this paper, we have performed some comparative experiments and provide some sample results in this paper.

The benefits of PageRank are the greatest for underspecified queries. For example, a query for "Stanford University" may return any number of web pages which mention Stanford (such as publication lists) on a conventional search engine, but using PageRank, the university home page is listed first.

5.1 Title Search

To test the usefulness of PageRank for search we implemented a search engine that used only the titles of 16 million web pages. To answer a query, the search engine finds all the web pages whose titles contain all of the query words. Then it sorts the results by PageRank. This search engine is very simple and cheap to implement. In informal tests, it worked remarkably well. As can be seen in Figure 6, a search for "University" yields a list of top universities. This figure shows our MultiQuery system which allows a user to query two search engines at the same time. The search engine on the left is our PageRank based title search engine. The bar graphs and percentages shown are a log of the actual PageRank with the top page normalized to 100%, not a percentile which is used everywhere else in this paper. The search engine on the right is Altavista. You can see that Altavista returns random looking web pages that match the query "University" and are the root page of the server (Altavista seems to be using URL length as a quality heuristic).

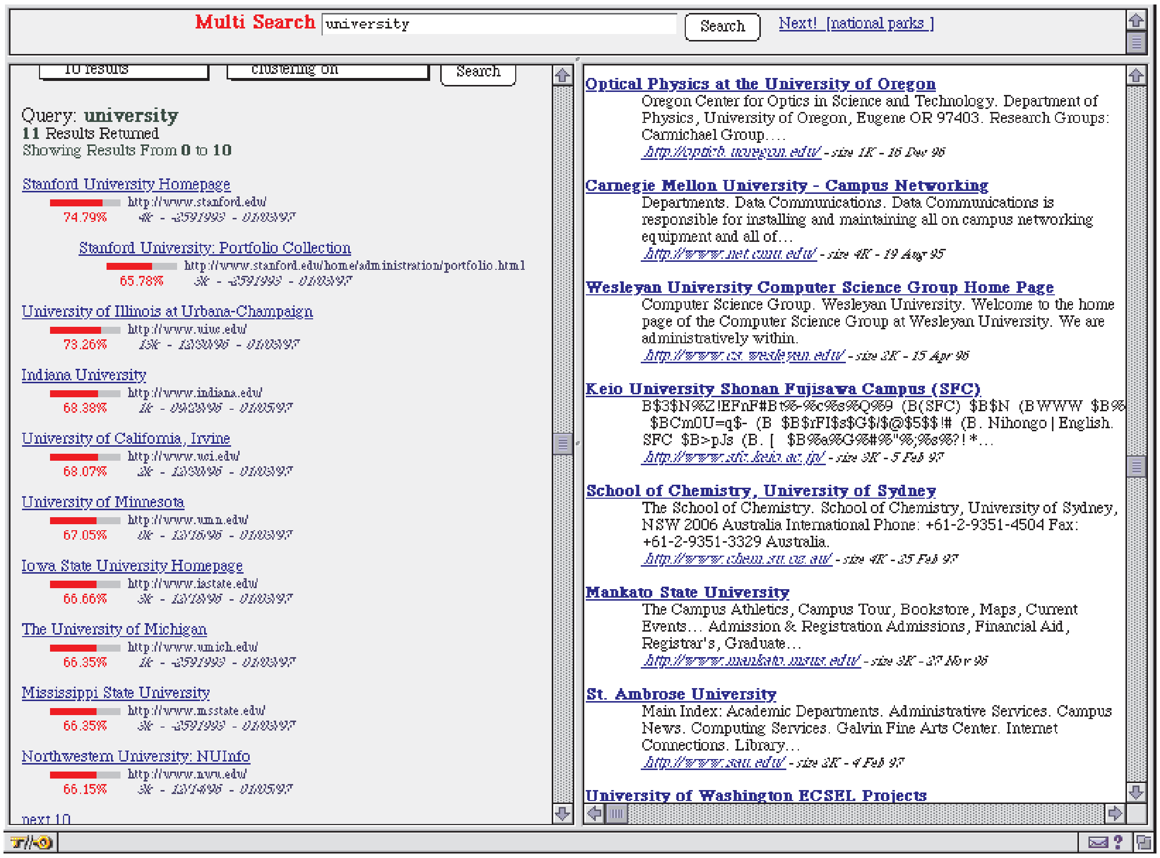

Figure 6: Comparison of Query for "University"

Web Page	PageRank (average is 1.0)
Download Netscape Software	11589.00
http://www.w3.org/	10717.70
Welcome to Netscape	8673.51
Point: It's What You're Searching For	7930.92
Web-Counter Home Page	7254.97
The Blue Ribbon Campaign for Online Free Speech	7010.39
CERN Welcome	6562.49
Yahoo!	6561.80
Welcome to Netscape	6203.47
Wusage 4.1: A Usage Statistics System For Web Servers	5963.27
The World Wide Web Consortium (W3C)	5672.21
Lycos, Inc. Home Page	4683.31
Starting Point	4501.98
Welcome to Magellan!	3866.82
Oracle Corporation	3587.63

Table 1: Top 15 Page Ranks: July 1996

5.2 Rank Merging

The reason that the title based PageRank system works so well is that the title match ensures high precision, and the PageRank ensures high quality. When matching a query like "University" on the web, recall is not very important because there is far more than a user can look at. For more specific searches where recall is more important, the traditional information retrieval scores over full-text and the PageRank should be combined. Our Google system does this type of rank merging. Rank merging is known to be a very difficult problem, and we need to spend considerable additional effort before we will be able to do a reasonable evaluation of these types of queries. However, we do believe that using PageRank as a factor in these queries is quite beneficial.

5.3 Some Sample Results

We have experimented considerably with Google, a full-text search engine which uses PageRank. While a full-scale user study is beyond the scope of this paper, we provide a sample query in Appendix A. For more queries, we encourage the reader to test Google themselves [BP].

Table 1 shows the top 15 pages based on PageRank. This particular listing was generated in July 1996. In a more recent calculation of PageRank, Microsoft has just edged out Netscape for the highest PageRank.

5.4 Common Case

One of the design goals of PageRank was to handle the common case for queries well. For example, a user searched for "wolverine", remembering that the University of Michigan system used for all administrative functions by students was called something with a wolverine in it. Our PageRank based title search system returned the answer "Wolverine Access" as the first result. This is sensible since all the students regularly use the Wolverine Access system, and a random user is quite likely to be looking for it given the query "wolverine". The fact that the Wolverine Access site is a good common case is not contained in the HTML of the page. Even if there were a way of defining good meta-information of this form within a page, it would be problematic since a page author could not be trusted with this kind of evaluation. Many web page authors would simply claim that their pages were all the best and most used on the web.

It is important to note that the goal of finding a site that contains a great deal of information about wolverines is a very different task than finding the common case wolverine site. There is an interesting system [Mar97] that attempts to find sites that discuss a topic in detail by propagating the textual matching score through the link structure of the web. It then tries to return the page on the most central path. This results in good results for queries like "flower"; the system will return good navigation pages from sites that deal with the topic of flowers in detail. Contrast that with the common case approach which might simply return a commonly used commercial site that had little information except how to buy flowers. It is our opinion that both of these tasks are important, and a general purpose web search engine should return results which fulfill the needs of both of these tasks automatically. In this paper, we are concentrating only on the common case approach.

5.5 Subcomponents of Common Case

It is instructive to consider what kind of common case scenarios PageRank can help represent. Besides a page which has a high usage, like the Wolverine Access cite, PageRank can also represent a collaborative notion of authority or trust. For example, a user might prefer a news story simply because it is linked is linked directly from the New York Times home page. Of course such a story will receive quite a high PageRank simply because it is mentioned by a very important page. This seems to capture a kind of collaborative trust, since if a page was mentioned by a trustworthy or authoritative source, it is more likely to be trustworthy or authoritative. Similarly, quality or importance seems to fit within this kind of circular definition.

6 Personalized PageRank

An important component of the PageRank calculation is E – a vector over the Web pages which is used as a source of rank to make up for the rank sinks such as cycles with no outedges (see Section 2.4). However, aside from solving the problem of rank sinks, E turns out to be a powerful parameter to adjust the page ranks. Intuitively the E vector corresponds to the distribution of web pages that a random surfer periodically jumps to. As we see below, it can be used to give broad general views of the Web or views which are focussed and personalized to a particular individual.

We have performed most experiments with an E vector that is uniform over all web pages with $||E||_1 = 0.15$. This corresponds to a random surfer periodically jumping to a random web page. This is a very democratic choice for E since all web pages are valued simply because they exist. Although this technique has been quite successful, there is an important problem with it. Some Web pages with many related links receive an overly high ranking. Examples of these include copyright warnings, disclaimers, and highly interlinked mailing list archives.

Another extreme is to have E consist entirely of a single web page. We tested two such E's – the Netscape home page, and the home page of a famous computer scientist, John McCarthy. For the Netscape home page, we attempt to generate page ranks from the perspective of a novice user who has Netscape set as the default home page. In the case of John McCarthy's home page we want to calculate page ranks from the perspective of an individual who has given us considerable contextual information based on the links on his home page.

In both cases, the mailing list problem mentioned above did not occur. And, in both cases, the respective home page got the highest PageRank and was followed by its immediate links. From

Web Page Title	John McCarthy's View PageRank Percentile	Netscape's View PageRank Percentile
John McCarthy's Home Page	100.00%	99.23%
John Mitchell (Stanford CS Theory Group)	100.00%	93.89%
Venture Law (Local Startup Law Firm)	99.94%	99.82%
Stanford CS Home Page	100.00%	99.83%
University of Michigan AI Lab	99.95%	99.94%
University of Toronto CS Department	99.99%	99.09%
Stanford CS Theory Group	99.99%	99.05%
Leadershape Institute	95.96%	97.10%

Table 2: Page Ranks for Two Different Views: Netscape vs. John McCarthy

that point, the disparity decreased. In Table 2, we show the resulting page rank percentiles for an assortment of different pages. Pages related to computer science have a higher McCarthy-rank than Netscape-rank and pages related to computer science at Stanford have a considerably higher McCarthy-rank. For example, the Web page of another Stanford Computer Science Dept. faculty member is more than six percentile points higher on the McCarthy-rank. Note that the page ranks are displayed as percentiles. This has the effect of compressing large differences in PageRank at the top of the range.

Such personalized page ranks may have a number of applications, including personal search engines. These search engines could save users a great deal of trouble by efficiently guessing a large part of their interests given simple input such as their bookmarks or home page. We show an example of this in Appendix A with the "Mitchell" query. In this example, we demonstrate that while there are many people on the web named Mitchell, the number one result is the home page of a colleague of John McCarthy named John Mitchell.

6.1 Manipulation by Commercial Interests

These types of personalized PageRanks are virtually immune to manipulation by commercial interests. For a page to get a high PageRank, it must convince an important page, or a lot of non-important pages to link to it. At worst, you can have manipulation in the form of buying advertisements(links) on important sites. But, this seems well under control since it costs money. This immunity to manipulation is an extremely important property. This kind of commercial manipulation is causing search engines a great deal of trouble, and making features that would be great to have very difficult to implement. For example fast updating of documents is a very desirable feature, but it is abused by people who want to manipulate the results of the search engine.

A compromise between the two extremes of uniform E and single page E is to let E consist of all the root level pages of all web servers. Notice this will allow some manipulation of PageRanks. Someone who wished to manipulate this system could simply create a large number of root level servers all pointing at a particular site.

7 Applications

7.1 Estimating Web Traffic

Because PageRank roughly corresponds to a random web surfer (see Section 2.5), it is interesting to see how PageRank corresponds to actual usage. We used the counts of web page accesses from NLANR [NLA] proxy cache and compared these to PageRank. The NLANR data was from several national proxy caches over the period of several months and consisted of 11,817,665 unique URLs with the highest hit count going to Altavista with 638,657 hits. There were 2.6 million pages in the intersection of the cache data and our 75 million URL database. It is extremely difficult to compare these datasets analytically for a number of different reasons. Many of the URLs in the cache access data are people reading their personal mail on free email services. Duplicate server names and page names are a serious problem. Incompleteness and bias a problem is both the PageRank data and the usage data. However, we did see some interesting trends in the data. There seems to be a high usage of pornographic sites in the cache data, but these sites generally had low PageRanks. We believe this is because people do not want to link to pornographic sites from their own web pages. Using this technique of looking for differences between PageRank and usage, it may be possible to find things that people like to look at, but do not want to mention on their web pages. There are some sites that have a very high usage, but low PageRank such as netscape.yahoo.com. We believe there is probably an important backlink which simply is omitted from our database (we only have a partial link structure of the web). It may be possible to use usage data as a start vector for PageRank, and then iterate PageRank a few times. This might allow filling in holes in the usage data. In any case, these types of comparisons are an interesting topic for future study.

7.2 PageRank as Backlink Predictor

One justification for PageRank is that it is a predictor for backlinks. In [CGMP98] we explore the issue of how to crawl the web efficiently, trying to crawl better documents first. We found on tests of the Stanford web that PageRank is a better predictor of future citation counts than citation counts themselves.

The experiment assumes that the system starts out with only a single URL and no other information, and the goal is to try to crawl the pages in as close to the optimal order as possible. The optimal order is to crawl pages in exactly the order of their rank according to an evaluation function. For the purposes here, the evaluation function is simply the number of citations, given complete information. The catch is that all the information to calculate the evaluation function is not available until after all the documents have been crawled. It turns out using the incomplete data, PageRank is a more effective way to order the crawling than the number of known citations. In other words, PageRank is a better predictor than citation counting even when the measure is the number of citations! The explanation for this seems to be that PageRank avoids the local maxima that citation counting gets stuck in. For example, citation counting tends to get stuck in local collections like the Stanford CS web pages, taking a long time to branch out and find highly cited pages in other areas. PageRank quickly finds the Stanford homepage is important, and gives preference to its children resulting in an efficient, broad search.

This ability of PageRank to predict citation counts is a powerful justification for using PageRank. Since it is very difficult to map the citation structure of the web completely, PageRank may even be a better citation count approximation than citation counts themselves.

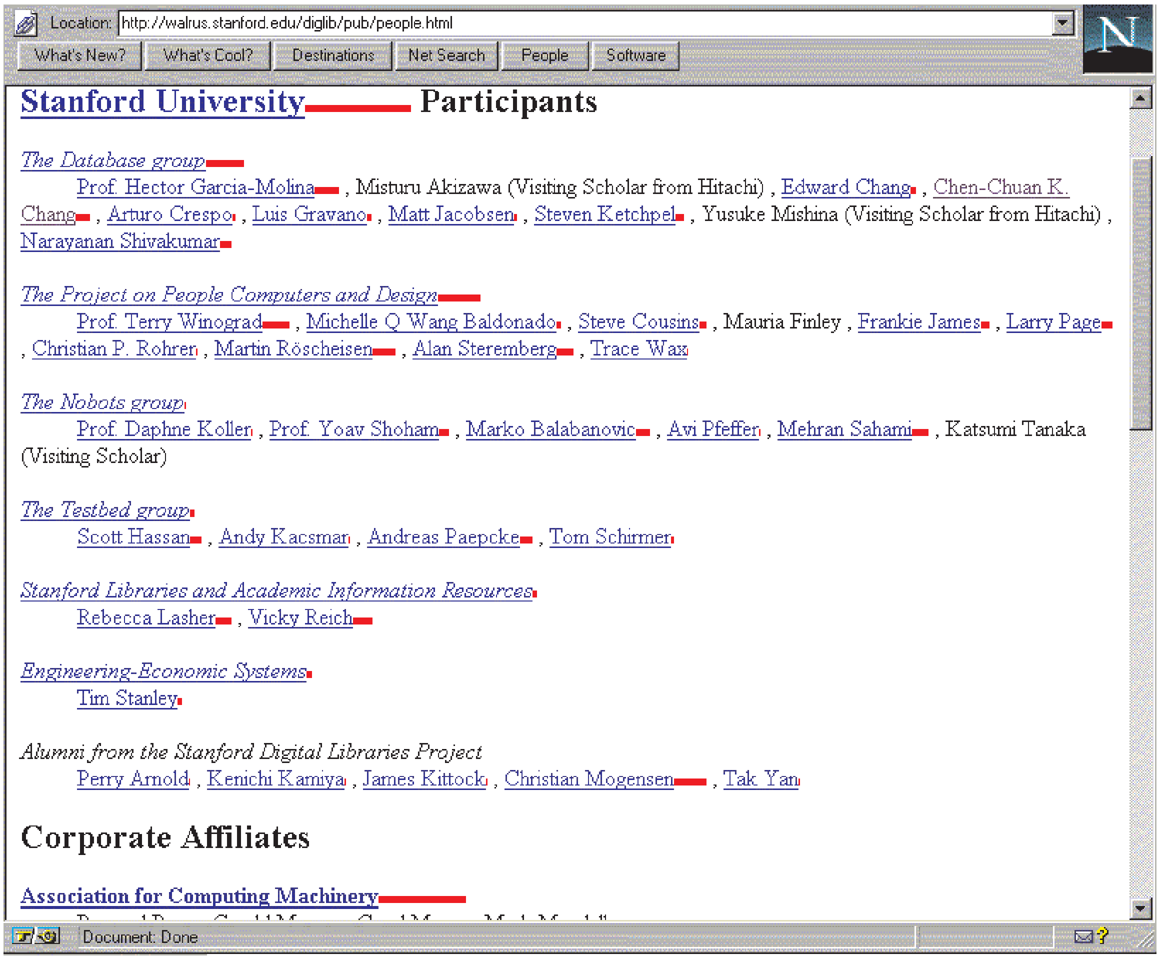

Figure 7: PageRank Proxy

7.3 User Navigation: The PageRank Proxy

We have developed a web proxy application that annotates each link that a user sees with its PageRank. This is quite useful, because users receive some information about the link before they click on it. In Figure 7 is a screen shot from the proxy. The length of the red bars is the log of the URL's PageRank. We can see that major organizations, like Stanford University, receive a very high ranking followed by research groups, and then people, with professors at the high end of the people scale. Also notice ACM has a very high PageRank, but not as high as Stanford University. Interestingly, this PageRank annotated view of the page makes an incorrect URL for one of the professors glaringly obvious since the professor has a embarrassingly low PageRank. Consequently this tool seems useful for authoring pages as well as navigation. This proxy is very helpful for looking at the results from other search engines, and pages with large numbers of links such as Yahoo's listings. The proxy can help users decide which links in a long listing are more likely to be interesting. Or, if the user has some idea where the link they are looking for should fall in the "importance" spectrum, they should be able to scan for it much more quickly using the proxy.

7.4 Other Uses of PageRank

The original goal of PageRank was a way to sort backlinks so if there were a large number of backlinks for a document, the "best" backlinks could be displayed first. We have implemented such a system. It turns out this view of the backlinks ordered by PageRank can be very interesting when trying to understand your competition. For example, the people who run a news site always want to keep track of any significant backlinks the competition has managed to get. Also, PageRank can help the user decide if a site is trustworthy or not. For example, a user might be inclined to trust information that is directly cited from the Stanford homepage.

8 Conclusion

In this paper, we have taken on the audacious task of condensing every page on the World Wide Web into a single number, its PageRank. PageRank is a global ranking of all web pages, regardless of their content, based solely on their location in the Web's graph structure.

Using PageRank, we are able to order search results so that more important and central Web pages are given preference. In experiments, this turns out to provide higher quality search results to users. Th intuition behind PageRank is that it uses information which is exernal to the Web pages themselves - their backlinks, which provide a kind of peer review. Furthermore, backlinks from "important" pages are more significant than backlinks from average pages. This is encompassed in the recursive definition of PageRank (Section 2.4).

PageRank could be used to separate out a small set of commonly used documents which can answer most queries. The full database only needs to be consulted when the small database is not adequate to answer a query. Finally, PageRank may be a good way to help find representative pages to display for a cluster center.

We have found a number of applications for PageRank in addition to search which include traffic estimation, and user navigation. Also, we can generate personalized PageRanks which can create a view of Web from a particular perspective.

Overall, our experiments with PageRank suggest that the structure of the Web graph is very useful for a variety of information retrieval tasks.

References

[BP] Sergey Brin and Larry Page. Google search engine. `http://google.stanford.edu`.

[CGMP98] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. Efficient crawling through url ordering. In *To Appear: Proceedings of the Seventh International Web Conference (WWW 98)*, 1998.

[Gar95] Eugene Garfield. New international professional society signals the maturing of scientometrics and informetrics. *The Scientist*, 9(16), Aug 1995. `http://www.the-scientist.library.upenn.edu/yr1995/august/issi_950821.ht%ml`.

[Gof71] William Goffman. A mathematical method for analyzing the growth of a scientific discipline. *Journal of the ACM*, 18(2):173–185, April 1971.

[Kle98] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the Nineth Annual ACM-SIAM Symposium on Discrete Algorithms.*, 1998.

[Mar97] Massimo Marchiori. The quest for correct information on the web: Hyper search engines. In *Proceedings of the Sixth International WWW Conference, Santa Claram USA, April, 1997*, 1997. `http://www6.nttlabs.com/HyperNews/get/PAPER222.html`.

[MF95] Sougata Mukherjea and James D. Foley. Showing the context of nodes in the world-wide web. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 2 of *Short Papers: Web Browsing*, pages 326–327, 1995.

[MFH95] Sougata Mukherjea, James D. Foley, and Scott Hudson. Visualizing complex hypermedia networks through multiple hierarchical views. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1 of *Papers: Creating Visualizations*, pages 331–337, 1995.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[NLA] NLANR. A distributed testbed for national information provisioning. `http://ircache.nlanr.net/Cache/`.

[PB98] Lawrence Page and Sergey Brin. The anatomy of a large-scale hypertextual web search engine. In *To Appear: Proceedings of the Seventh International Web Conference (WWW 98)*, 1998.

[Pit97] James E. Pitkow. *Characterizing World Wide Web Ecologies*. PhD thesis, Georgia Institue of Technology, June 1997.

[PPR96] Peter Pirolli, James Pitkow, and Ramana Rao. Silk from a sow's ear: Extracting usable structure from the web. In Michael J. Tauber, Victoria Bellotti, Robin Jeffries, Jock D. Mackinlay, and Jakob Nielsen, editors, *Proceedings of the Conference on Human Factors in Computing Systems : Common Ground*, pages 118–125, New York, 13–18 April 1996. ACM Press.

[San95] Neeraja Sankaran. Speculation in the biomedical community abounds over likely candidates for nobel. *The Scientist*, 9(19), Oct 1995. `http://www.the-scientist.library.upenn.edu/yr1995/oct/nobel_951002.html%`.

[Spe97] Ellen Spertus. Parasite: Mining structural information on the web. In *Proceedings of the Sixth International WWW Conference, Santa Claram USA, April, 1997*, 1997. `http://www6.nttlabs.com/HyperNews/get/PAPER206.html`.

[Til] Hope N. Tillman. Evaluating quality on the net. `http://www.tiac.net/users/hope/findqual.html`.

[WVS+96] Ron Weiss, Bienvenido Vélez, Mark A. Sheldon, Chanathip Manprempre, Peter Szilagyi, Andrzej Duda, and David K. Gifford. HyPursuit: A hierarchical network search engine that exploits content-link hypertext clustering. In *Proceedings of the 7th ACM Conference on Hypertext*, pages 180–193, New York, 16–20 March 1996. ACM Press.

Appendix A

This Appendix contains several sample query results using Google. The first is a query for “Digital Libraries” using a uniform E. The next is a query for “Mitchell” using E consisting just of John McCarthy’s home page.