Computer Networks 33 (2000) 309–320
COMPUTER NETWORKS
www.elsevier.com/locate/comnet

Graph structure in the Web

Andrei Broder [a], Ravi Kumar [b,*], Farzin Maghoul [a], Prabhakar Raghavan [b], Sridhar Rajagopalan [b], Raymie Stata [c], Andrew Tomkins [b], Janet Wiener [c]

[a] *AltaVista Company, San Mateo, CA, USA*
[b] *IBM Almaden Research Center, San Jose, CA, USA*
[c] *Compaq Systems Research Center, Palo Alto, CA, USA*

Abstract

The study of the Web as a graph is not only fascinating in its own right, but also yields valuable insight into Web algorithms for crawling, searching and community discovery, and the sociological phenomena which characterize its evolution. We report on experiments on local and global properties of the Web graph using two AltaVista crawls each with over 200 million pages and 1.5 billion links. Our study indicates that the macroscopic structure of the Web is considerably more intricate than suggested by earlier experiments on a smaller scale.

Keywords: Graph structure; Diameter; Web measurement

1. Introduction

Consider the directed graph whose nodes correspond to static pages on the Web, and whose arcs correspond to links between these pages. We study various properties of this graph including its diameter, degree distributions, connected components, and macroscopic structure. There are several reasons for developing an understanding of this graph.

(1) Designing crawl strategies on the Web [15].
(2) Understanding of the sociology of content creation on the Web.
(3) Analyzing the behavior of Web algorithms that make use of link information [9–11,20,26]. To take just one example, what can be said of the distribution and evolution of PageRank [9] values on graphs like the Web?
(4) Predicting the evolution of Web structures such as bipartite cores [21] and Webrings, and developing better algorithms for discovering and organizing them.
(5) Predicting the emergence of important new phenomena in the Web graph.

We detail a number of experiments on a Web crawl of approximately 200 million pages and 1.5 billion links; the scale of this experiment is thus five times larger than the previous biggest study [21] of structural properties of the Web graph, which used a pruned data set from 1997 containing about 40 million pages. Recent work ([21] on the 1997 crawl, and [5] on the approximately 325 thousand node nd.edu subset of the Web) has suggested that the distribution of degrees (especially in-degrees, i.e., the number of links to a page) follows a *power law*.

The power law for in-degree: the probability

* Corresponding author. E-mail: ravi@almaden.ibm.com

1389-1286/00/$ – see front matter
PII: S1389-1286(00)00083-9

that a node has in-degree i is proportional to $1/i^x$, for some $x > 1$.

We verify the power law phenomenon in current (considerably larger) Web crawls, confirming it as a basic Web property.

In other recent work, [4] report the intriguing finding that most pairs of pages on the Web are separated by a handful of links, almost always under 20, and suggest that this number will grow logarithmically with the size of the Web. This is viewed by some as a 'small world' phenomenon. Our experimental evidence reveals a rather more detailed and subtle picture: most ordered pairs of pages cannot be bridged at all and there are significant numbers of pairs that can be bridged, but only using paths going through hundreds of intermediate pages. Thus, the Web is not the ball of highly connected spaghetti we believed it to be; rather, the connectivity is strongly limited by a high-level global structure.

1.1. Our main results

We performed three sets of experiments on Web crawls from May 1999 and October 1999. Unless otherwise stated, all results described below are for the May 1999 crawl, but all conclusions have been validated for the October 1999 crawl as well. First, we generated the in- and out-degree distributions, confirming previous reports on power laws; for instance, the fraction of Web pages with i in-links is proportional to $1/i^{2.1}$. The constant 2.1 is in remarkable agreement with earlier studies at varying scales [5,21]. In our second set of experiments we studied the directed and undirected connected components of the Web. We show that power laws also arise in the distribution of *sizes* of these connected components. Finally, in our third set of experiments, we performed a number of breadth-first searches from randomly chosen start nodes. We detail these experiments in Section 2.

Our analysis reveals an interesting picture (Fig. 9) of the Web's macroscopic structure. Most (over 90%) of the approximately 203 million nodes in our May 1999 crawl form a single connected component if links are treated as *undirected* edges. This connected Web breaks naturally into four pieces. The first piece is a central core, all of whose pages can reach one another along directed links; this 'giant strongly connected component' (*SCC*) is at the heart of the Web. The second and third pieces are called *IN* and *OUT*. *IN* consists of pages that can reach the *SCC*, but cannot be reached from it; possibly new sites that people have not yet discovered and linked to. *OUT* consists of pages that are accessible from the *SCC*, but do not link back to it, such as corporate Websites that contain only internal links. Finally, the *TENDRILS* contain pages that cannot reach the *SCC*, and cannot be reached from the *SCC*. Perhaps the most surprising fact is that the size of the *SCC* is relatively small; it comprises about 56 million pages. Each of the other three sets contain about 44 million pages, thus, all four sets have roughly the same size.

We show that the diameter of the central core (*SCC*) is at least 28, and that the diameter of the graph as a whole is over 500 (see Section 1.3 for definitions of diameter). We show that for randomly chosen source and destination pages, the probability that any path exists from the source to the destination is only 24%. We also show that, if a directed path exists, its average length will be about 16. Likewise, if an undirected path exists (i.e., links can be followed forwards or backwards), its average length will be about 6. These analyses appear in the Section 3. These results are remarkably consistent across two different, large AltaVista crawls. This suggests that our results are relatively insensitive to the particular crawl we use, provided it is large enough. We will say more about crawl effects in Section 3.4.

In a sense the Web is much like a complicated organism, in which the local structure at a microscopic scale looks very regular like a biological cell, but the global structure exhibits interesting morphological structure (body and limbs) that are not obviously evident in the local structure. Therefore, while it might be tempting to draw conclusions about the structure of the Web graph from a local picture of it, such conclusions may be misleading.

1.2. Related prior work

Broadly speaking, related prior work can be classified into two groups: (1) observations of the power law distributions on the Web; and (2) work on applying graph theoretic methods to the Web.

1.2.1. Zipf–Pareto–Yule and power laws

Distributions with an inverse polynomial tail have been observed in a number of contexts. The earliest observations are due to Pareto [27] in the context of economic models. Subsequently, these statistical behaviors have been observed in the context of literary vocabulary [32], sociological models [33], and even oligonucleotide sequences [24] among others. Our focus is on the closely related power law distributions, defined on the positive integers, with the probability of the value i being proportional to $1/i^k$ for a small positive number k. Perhaps the first rigorous effort to define and analyze a model for power law distributions is due to Simon [30].

More recently, power law distributions have been observed in various aspects of the Web. Two lines of work are of particular interest to us. First, power laws have been found to characterize user behavior on the Web in two related but dual forms:

(1) access statistics for Web pages, which can be easily obtained from server logs (but for caching effects); see [1,2,17,19];

(2) numbers of times users at a single site access particular pages, as verified by instrumenting and inspecting logs from Web caches, proxies, and clients (see [6] and references therein, as well as [23]).

Second, and more relevant to our immediate context, is the distribution of degrees on the Web graph. In this context, recent work (see [5,21]) suggests that both the in- and the out-degrees of vertices on the Web graph have power laws. The difference in scope in these two experiments is noteworthy. The first [21] examines a Web crawl from 1997 due to Alexa, Inc., with a total of over 40 million nodes. The second [5], examines Web pages from the University of Notre Dame domain, *.nd.edu, as well as a portion of the Web reachable from three other URLs. In this paper, we verify these power laws on more recent (and considerably larger) Web crawls. This collection of findings reveals an almost fractal-like quality for the power law in-degree and out-degree distributions, in that it appears both as a macroscopic phenomenon on the entire Web, as a microscopic phenomenon at the level of a single university Website, and at intermediate levels between these two.

There is no evidence that users' browsing behavior, access statistics and the linkage statistics on the Web graph are related in any fundamental way, although it is very tempting to conjecture that this is indeed the case. It is usually the case, though not always so, that pages with high in-degree will also have high PageRank [9]. Indeed, one way of viewing PageRank is that it puts a number on how easy (or difficult) it is to find particular pages by a browsing-like activity. Consequently, it is plausible that the in-degree distributions induce a similar distribution on browsing activity and consequently, on access statistics.

Faloutsos et al. [16] observe Zipf–Pareto distributions (power law distributions on the *ranks* of values) on the Internet network topology using a graph of the network obtained from the routing tables of a backbone BGP router.

1.2.2. Graph theoretic methods

Much recent work has addressed the Web as a graph and applied algorithmic methods from graph theory in addressing a slew of search, retrieval, and mining problems on the Web. The efficacy of these methods was already evident even in early local expansion techniques [10]. Since then, the increasing sophistication of the techniques used, the incorporation of graph theoretical methods with both classical and new methods which examine context and content, and richer browsing paradigms have enhanced and validated the study and use of such methods. Following Butafogo and Schneiderman [10], the view that connected and strongly connected components represent meaningful entities has become accepted. Pirolli et al. [28] augment graph theoretic analysis to include document content, as well as usage statistics, resulting in a rich understanding of domain structure and a taxonomy of roles played by Web pages.

Graph theoretic methods have been used for search [8,9,12,13,20], browsing and information foraging [10,11,14,28,29], and Web mining [21,22,25,26]. We expect that a better structural characterization of the Web will have much to say in each of these contexts.

In this section we formalize our view of the Web as a graph; in this view we ignore the text and other content in pages, focusing instead on the links between pages. Adopting the terminology of graph theory [18], we refer to pages as *nodes*, and to links as *arcs*. In this framework, the Web becomes a large graph containing several hundred million nodes, and a few billion arcs. We will refer to this graph as the *Web graph*, and our goal in this paper is to understand some of its properties. Before presenting our model for Web-like graphs, we begin with a brief primer on graph theory, and a discussion of graph models in general.

1.3. A brief primer on graphs and terminology

The reader familiar with basic notions from graph theory may skip this primer.

A *directed graph* consists of a set of *nodes,* denoted V and a set of *arcs,* denoted E. Each arc is an ordered pair of nodes (u, v) representing a directed connection from u to v. The *out-degree* of a node u is the number of distinct arcs $(u, v_1)\ldots(u, v_k)$ (i.e., the number of links from u), and the *in-degree* is the number of distinct arcs $(v_1, u)\ldots(v_k, u)$ (i.e., the number of links to u). A path from node u to node v is a sequence of arcs $(u, u_1), (u_1, u_2), \ldots(u_k, v)$. One can follow such a sequence of arcs to 'walk' through the graph from u to v. Note that a path from u to v does not imply a path from v to u. The *distance* from u to v is one more than the smallest k for which such a path exists. If no path exists, the distance from u to v is defined to be infinity. If (u, v) is an arc, then the distance from u to v is 1.

Given a directed graph, a *strongly connected component* (strong component for brevity) of this graph is a set of nodes such that for any pair of nodes u and v in the set there is a path from u to v. In general, a directed graph may have one or many strong components. The strong components of a graph consist of disjoint sets of nodes. One focus of our studies will be in understanding the distribution of the sizes of strong components on the Web graph.

An *undirected graph* consists of a set of nodes and a set of *edges*, each of which is an unordered pair $\{u, v\}$ of nodes. In our context, we say there is an edge between u and v if there is a link between u and v, without regard to whether the link points from u to v or the other way around. The *degree* of a node u is the number of edges incident to u. A path is defined as for directed graphs, except that now the existence of a path from u to v implies a path from v to u. A *component* of an undirected graph is a set of nodes such that for any pair of nodes u and v in the set there is a path from u to v. We refer to the components of the undirected graph obtained from a directed graph by ignoring the directions of its arcs as the *weak components* of the directed graph. Thus two nodes on the Web may be in the same weak component even though there is no *directed* path between them (consider, for instance, a node u that points to two other nodes v and w; then v and w are in the same weak component even though there may be no sequence of links leading from v to w or vice versa). The interplay of strong and weak components on the (directed) Web graph turns out to reveal some unexpected properties of the Web's connectivity.

A *breadth-first search* (BFS) on a directed graph begins at a node u of the graph, and proceeds to build up the set of nodes reachable from u in a series of layers. Layer 1 consists of all nodes that are pointed to by an arc from u. Layer k consists of all nodes to which there is an arc from some vertex in layer $k - 1$, but are not in any earlier layer. Notice that by definition, layers are disjoint. The distance of any node from u can be read out of the breadth-first search. The shortest path from u to v is the index of the layer v belongs in, i.e., if there is such a layer. On the other hand, note that a node that cannot be reached from u does not belong in any layer, and thus we define the distance to be infinity. A BFS on an undirected graph is defined analogously.

Finally, we must take a moment to describe the exact notions of diameter we study, since several have been discussed informally in the context of the Web. Traditionally, the *diameter* of a graph, directed or undirected, is the maximum over all ordered pairs (u, v) of the shortest path from u to v. Some researchers have proposed studying the *average distance* of a graph, defined to be the length of the shortest path from u to v, averaged over all ordered pairs (u, v); this is referred to as diameter in [4]. The difficulty with this notion is that even a single pair (u, v) with no path from u to v results in an infinite average distance. In fact, as we show from our experiments below, the Web is rife with such pairs (thus it is not merely a matter of discarding a few outliers before taking this average). This motivates the following revised definition: let P be the set of all ordered pairs (u, v) such that there is a path from u to v. The *average connected distance* is the expected length of the shortest path, where the expectation is over uniform choices from P.

2. Experiments and results

2.1. Infrastructure

All experiments were run using the Connectivity Server 2 (CS2) software built at Compaq Systems Research Center using data provided by AltaVista. CS2 provides fast access to linkage information on the Web. A build of CS2 takes a Web crawl as input and creates a representation of the entire Web graph induced by the pages in the crawl, in the form of a database that consists of all URLs that were crawled together with all in-links and out-links among those URLs. In addition, the graph is extended with those URLs referenced at least five times by the crawled pages. (Experimentally, we have determined that the vast majority of URLs encountered fewer than five times but not crawled turn out to be invalid URLs.)

CS2 improves on the original connectivity server (CS1) described in [7] in two important ways. First, it significantly increases the compression of the URLs and the links to data structures. In CS1, each compressed URL is, on average, 16 bytes. In CS2, each URL is stored in 10 bytes. In CS1, each link requires 8 bytes to store as both an in-link and out-link; in CS2, an average of only 3.4 bytes are used. Second, CS2 provides additional functionality in the form of a host database. For example, in CS2, it is easy to get all the in-links for a given node, or just the in-links from remote hosts.

Like CS1, CS2 is designed to give high-performance access to all this data on a high-end machine with enough RAM to store the database in memory. On a 465 MHz Compaq AlphaServer 4100 with 12 GB of RAM, it takes 70–80 μs to convert an URL into an internal id or vice versa, and then only 0.15 μs/link to retrieve each in-link or out-link. On a uniprocessor machine, a BFS that reaches 100 million nodes takes about 4 minutes; on a 2-processor machine we were able complete a BFS every 2 minutes.

In the experiments reported in this paper, CS2 was built from a crawl performed at AltaVista in May, 1999. The CS2 database contains 203 million URLs and 1466 million links (all of which fit in 9.5 GB of storage). Some of our experiments were repeated on a more recent crawl from October, 1999 containing 271 million URLs and 2130 million links.

In general, the AltaVista crawl is based on a large set of starting points accumulated over time from various sources, including voluntary submissions. The crawl proceeds in roughly a BFS manner, but is subject to various rules designed to avoid overloading Web servers, avoid robot traps (artificial infinite paths), avoid and/or detect spam (page flooding), deal with connection time outs, etc. Each build of the AltaVista index is based on the crawl data after further filtering and processing designed to remove duplicates and near duplicates, eliminate spam pages, etc. Then the index evolves continuously as various processes delete dead links, add new pages, update pages, etc. The secondary filtering and the later deletions and additions are not reflected in the connectivity server. But overall, CS2's database can be viewed as a superset of all pages stored in the index at one point in time. Note that due to the multiple starting points, it is possible for the resulting graph to have many connected components.

2.2. Experimental data

The following basic algorithms were implemented using CS2: (1) a BFS algorithm that performs a breadth-first traversal; (2) a WCC algorithm that finds the weak components; and (3) an SCC algorithm that finds the strongly connected components. Recall that both WCC and SCC are simple generalizations of the BFS algorithm. Using these three basic algorithms, we ran several interesting experiments on the Web graph.

2.2.1. Degree distributions

The first experiment we ran was to verify earlier observations that the in- and out-degree distributions on the Web are distributed according to power laws. We ran the experiment on both the May and October crawls of the Web. The results, shown in Figs. 1 and 3, show remarkable agreement with each other, and with similar experiments from data that is over two years old [21]. Indeed, in the case of in-degree, the exponent of the power law is consistently around 2.1, a number reported in [5,21]. The anomalous bump at 120 on the x-axis is due to a large clique formed by a single spammer. In all our log–log plots, straight lines are linear regressions for the best power law fit.

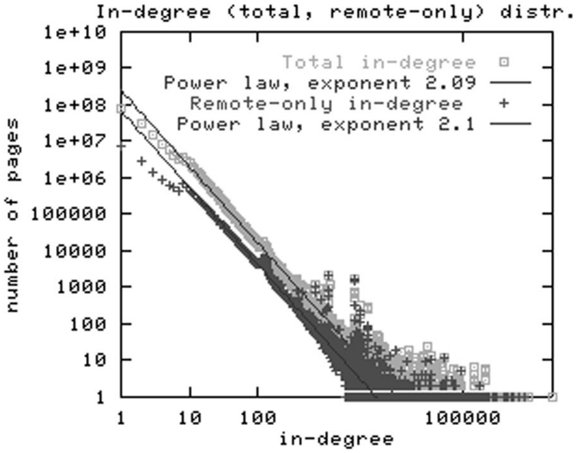

Fig. 1. In-degree distributions subscribe to the power law. The law also holds if only off-site (or 'remote-only') edges are considered.

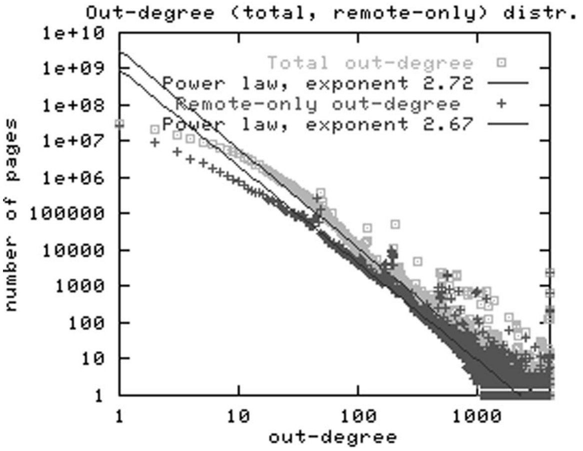

Fig. 2. Out-degree distributions subscribe to the power law. The law also holds if only off-site (or 'remote-only') edges are considered.

Out-degree distributions also exhibit a power law, although the exponent is 2.72, as can be seen in Figs. 2 and 4. It is interesting to note that the initial segment of the out-degree distribution deviates significantly from the power law, suggesting that pages with low out-degree follow a different (possibly Poisson or a combination of Poisson and power law, as suggested by the concavity of the deviation) distribution. Further research is needed to understand this combination better.

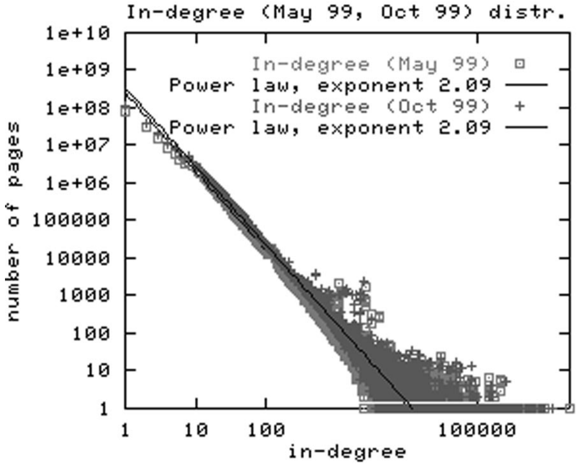

Fig. 3. In-degree distributions show a remarkable similarity over two crawls, run in May and October 1999. Each crawl counts well over 1 billion distinct edges of the Web graph.

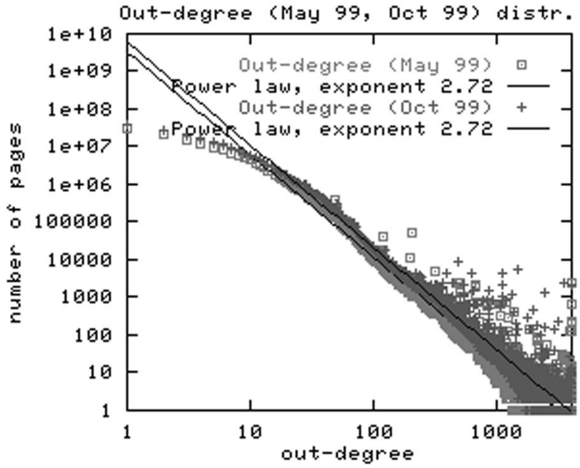

Fig. 4. Out-degree distributions show a remarkable similarity over two crawls, run in May and October 1999. Each crawl counts well over 1 billion distinct edges of the Web graph.

2.2.2. Undirected connected components

In the next set of experiments we treat the Web graph as an undirected graph and find the sizes of the undirected components. We find a giant component of 186 million nodes in which fully 91% of the nodes in our crawl are reachable from one another by following either forward or backward links. This is done by running the WCC algorithm which simply finds all connected components in the undirected Web graph. Thus, if one could browse along both

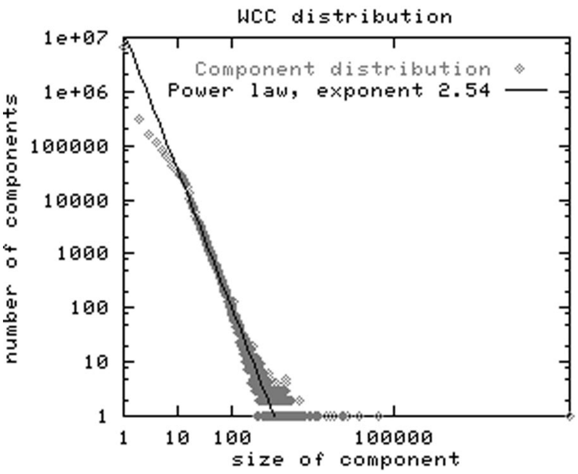

Fig. 5. Distribution of weakly connected components on the Web. The sizes of these components also follow a power law.

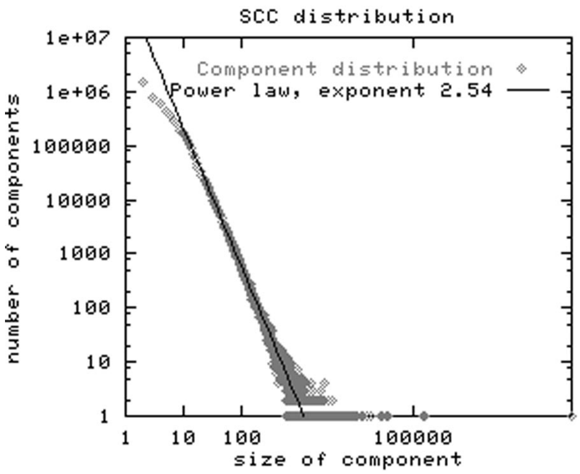

Fig. 6. Distribution of strongly connected components on the Web. The sizes of these components also follow a power law.

forward and backward directed links, the Web is a very well connected graph. Surprisingly, even the distribution of the sizes of WCCs exhibits a power law with exponent roughly 2.5 (Fig. 5).

Does this widespread connectivity result from a few nodes of large in-degree acting as ‘junctions’? Surprisingly, this turns out not to be the case. Indeed, even if all links to pages with in-degree 5 or higher are removed (certainly including links to every well-known page on the Web), the graph still contains a giant weak component of size 59 million (see Table 1). This provides us with two interesting and useful insights. First, the connectivity of the Web graph as an undirected graph is extremely resilient and does not depend on the existence of nodes of high in-degree. Second, such nodes, which are very useful and tend to include nodes with high PageRank or nodes that are considered good hubs and authorities, are embedded in a graph that is well connected without them. This last fact may help understand why algorithms such as HITS [20] converge quickly.

Table 1
Size of the largest surviving weak component when links to pages with in-degree at least k are removed from the graph.

k	1000	100	10	5	4	3
Size (millions)	177	167	105	59	41	15

2.2.3. Strongly connected components

Motivated in part by the intriguing prediction of [4] that the average distance (referred to in their paper as diameter) of the Web is 19 (and thus it should be possible to get from any page to any other in a small number of clicks), we turned to the strongly connected components of the Web as a directed graph. By running the strongly connected component algorithm, we find that there is a single large SCC consisting of about 56 million pages, all other components are significantly smaller in size. This amounts to barely 28% of all the pages in our crawl. One may now ask: where have all the other pages gone? The answer to this question reveals some fascinating detailed structure in the Web graph; to expose this and to further study the issues of the diameter and average distance, we conducted a further series of experiments. Note that the distribution of the sizes of SCCs also obeys a power law (Fig. 6).

2.2.4. Random-start BFS

We ran the BFS algorithm twice from each of 570 randomly chosen starting nodes: once in the *forward* direction, following arcs of the Web graph as a browser would, and once *backward* following links in the reverse direction. Each of these BFS traversals (whether forward or backward) exhibited a sharp bimodal behavior: it would either ‘die out’ after reaching a small set of nodes (90% of the time this set has fewer than 90 nodes; in extreme cases it has a few hundred thousand), or it would ‘explode’ to cover about 100 million nodes (but never the entire 186 million). Further, for a fraction of the starting nodes, both the forward and the backward BFS runs would ‘explode’, each covering about 100 million nodes (though not the same 100 million in the two runs). As we show below, these are the starting points that lie in the SCC.

The cumulative distributions of the nodes covered in these BFS runs are summarized in Fig. 7. They reveal that the true structure of the Web graph must be somewhat subtler than a ‘small world’ phenomenon in which a browser can pass from any Web page to any other with a few clicks. We explicate this structure in Section 3.

2.2.5. Zipf distributions vs power law distributions

The *Zipf distribution* is an inverse polynomial function of *ranks* rather than magnitudes; for example, if only in-degrees 1, 4, and 5 occurred then a power law would be inversely polynomial in those values, whereas a Zipf distribution would be inversely polynomial in the ranks of those values: i.e., inversely polynomial in 1, 2, and 3. The in-degree distribution in our data shows a striking fit with a Zipf (more so than the power law) distribution; Fig. 8 shows the in-degrees of pages from the May 1999 crawl plotted against both ranks and magnitudes (corresponding to the Zipf and power law cases). The plot against ranks is virtually a straight line in the log–log plot, without the flare-out noticeable in the plot against magnitudes.

3. Interpretation and further work

Let us now put together the results of the connected component experiments with the results of the random-start BFS experiments. Given that the set SCC contains only 56 million of the 186 million nodes in our giant weak component, we use the BFS runs to estimate the positions of the remaining nodes. The

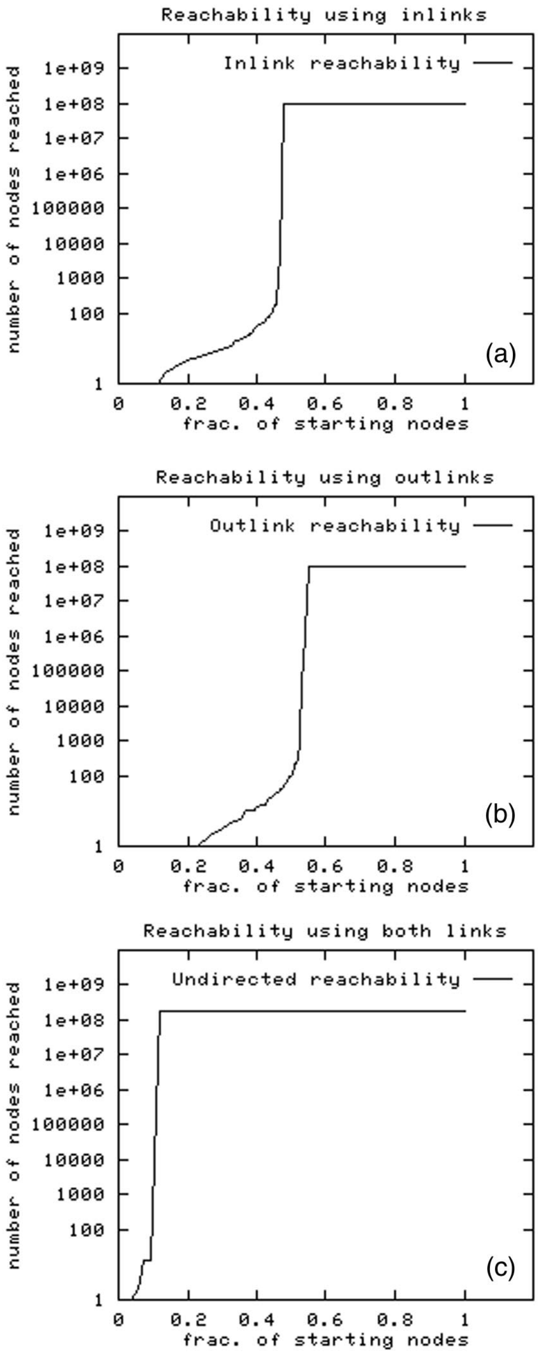

Fig. 7. Cumulative distribution on the number of nodes reached when BFS is started from a random node: (a) follows in-links, (b) follows out-links, and (c) follows both in- and out-links. Notice that there are two distinct regions of growth, one at the beginning and an ‘explosion’ in 50% of the start nodes in the case of in- and out-links, and for 90% of the nodes in the undirected case. These experiments form the basis of our structural analysis.

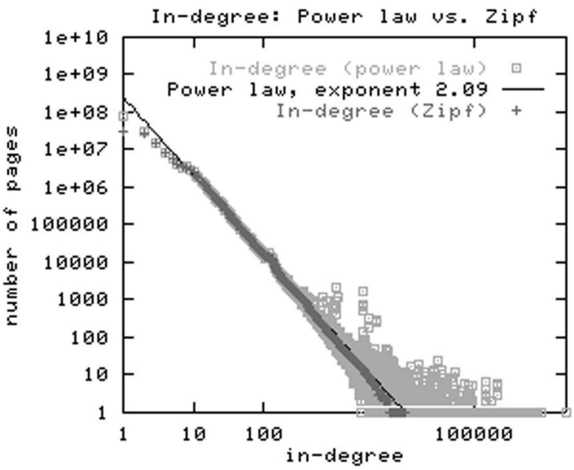

Fig. 8. In-degree distributions plotted as a power law and as a Zipf distribution.

starting points for which the forward BFS 'explodes' are either in SCC, or in a set we call IN, that has the following property: there is a directed path from each node of IN to (all the nodes of) SCC. Symmetrically, there is a set we call OUT containing all starting points for which the backward BFS 'explodes'; there is a directed path from any node in the SCC to every node in OUT. Thus a forward BFS from any node in either the SCC or IN will explode, as will a backward BFS from any node in either the SCC or OUT. By analyzing forward and backward BFS from 570 random starting points, we can compute the number of nodes that are in SCC, IN, OUT or none of these. Fig. 9 shows the situation as we can now infer it.

We now give a more detailed description of the structure in Fig. 9. The sizes of the various components are as follows:

Region	SCC	IN	OUT
Size	56,463,993	43,343,168	43,166,185
Region	TENDRILS	DISC.	Total
Size	43,797,944	16,777,756	203,549,046

These sizes were determined as follows. We know the total number of nodes in our crawl, so by subtracting the size of the giant weak component we determine the size of DISCONNECTED. Then our strong-component algorithm gives us the size of SCC. We turn to our breadth-first search data. As noted, searching from a particular start node following a particular type of edges (in-edges or out-edges) would either terminate quickly, or grow the search to about 100 million nodes. We say that a node *explodes* if it falls into the latter group. Thus, if a node explodes following in-links, and also explodes following out-links, it must be a member of a strong component of size at least $100 + 100 - 186 = 14$ million. Since the second largest strong component is of size 150 thousand, we infer that SCC is the unique strong component that contains all nodes exploding following in- as well as out-links. In fact, this observation contains two corroborating pieces of evidence for the structure in the table above: first, it turns out that the fraction of our randomly chosen BFS start nodes that explode under in- and out-links is the same as the fraction of nodes in the SCC as returned by our SCC algorithm. Second, every BFS start node in the SCC reaches exactly the same number of nodes under in-link expansion; this number is 99,807,161. Likewise, under out-link expansion every node of SCC reaches exactly 99,630,178 nodes.

Thus, we know that SCC + IN = 99,807,161, and similarly SCC + OUT = 99,630,178. Having already found the size of SCC, we can solve for IN and OUT. Finally, since we know the size of the giant weak component, we can subtract SCC, IN, and OUT to get TENDRILS. We now discuss each region in turn.

3.1. TENDRILS and DISCONNECTED

We had 172 samples from TENDRILS and DISCONNECTED; our BFS measurements cannot be used to differentiate between these two regions. By following out-links from a start point in this region, we encounter an average of 20 nodes before the exploration stops. Likewise, by following in-links we encounter an average of 52 nodes.

3.2. IN and OUT

Our sample contains 128 nodes from IN and 134 from OUT. We ask: when following out-links from nodes in OUT, or in-links from nodes in IN, how many nodes do we encounter before the BFS terminates? That is, how large a neighborhood do points in these

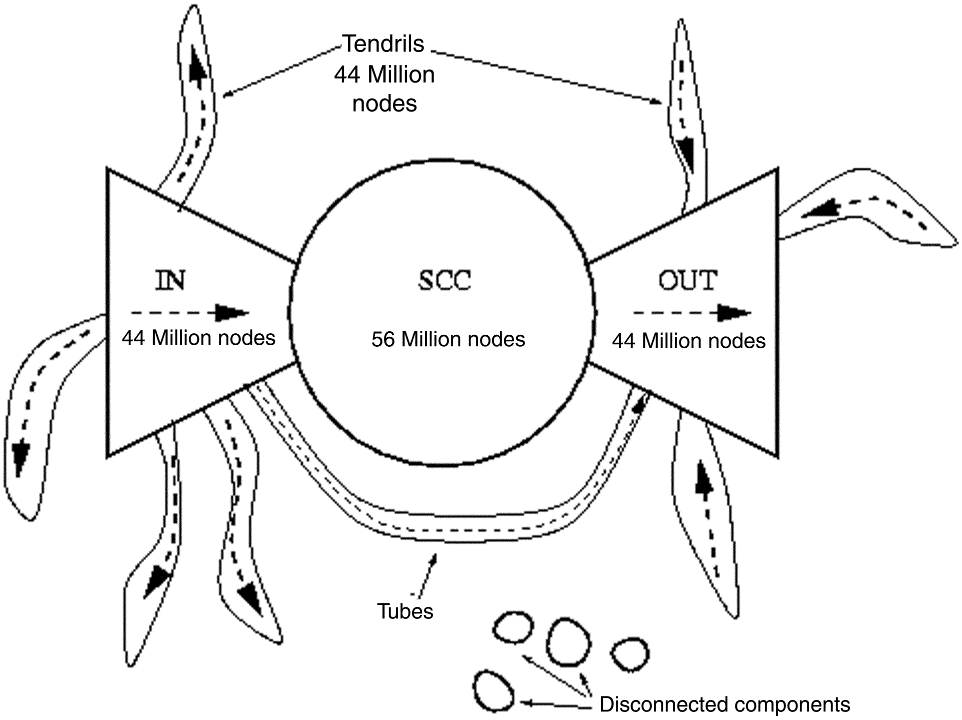

Fig. 9. Connectivity of the Web: one can pass from any node of IN through SCC to any node of OUT. Hanging off IN and OUT are TENDRILS containing nodes that are reachable from portions of IN, or that can reach portions of OUT, without passage through SCC. It is possible for a TENDRIL hanging off from IN to be hooked into a TENDRIL leading into OUT, forming a TUBE: i.e., a passage from a portion of IN to a portion of OUT without touching SCC.

regions have, if we explore in the direction ‘away’ from the center? The results are shown below in the row labeled ‘exploring outward – all nodes’.

Similarly, we know that if we explore in-links from a node in OUT, or out-links from a node in IN, we will encounter about 100 million other nodes in the BFS. Nonetheless, it is reasonable to ask: how many other nodes will we encounter? That is, starting from OUT (or IN), and following in-links (or out-links), how many nodes of TENDRILS and OUT (or IN) will we encounter? The results are shown below in the row labeled ‘exploring inwards – unexpected nodes’. Note that the numbers in the table represent averages over our sample nodes.

Starting point	OUT	IN
Exploring outwards – all nodes	3093	171
Exploring inwards – unexpected nodes	3367	173

As the table shows, OUT tends to encounter larger neighborhoods. For example, the second largest strong component in the graph has size approximately 150 thousand, and two nodes of OUT encounter neighborhoods a few nodes larger than this, suggesting that this component lies within OUT. In fact, considering that (for instance) almost every corporate Website not appearing in SCC will appear in OUT, it is no surprise that the neighborhood sizes are larger.

3.3. SCC

Our sample contains 136 nodes from the SCC. To determine other properties of SCC, we require a useful property of IN and OUT: each contains a few long paths such that, once the BFS proceeds beyond a certain depth, only a few paths are being explored, and the last path is much longer than any of the others. We can therefore explore the radius at which the BFS completes, confident that the last long path will be the same no matter which node of SCC we start from. The following table shows the depth at which the BFS terminates in each direction (following in-links or out-links) for nodes in the SCC.

Measure	Minimum depth	Average depth	Maximum depth
In-links	475	482	503
Out-links	430	434	444

As the table shows, from some nodes in the SCC it is possible to complete the search at distance 475, while from other nodes distance 503 is required. This allows us to conclude that the directed diameter of SCC is at least 28.

3.4. Other observations

As noted above, the (min, average, max) depths at which the BFS from SCC terminates following in-links are (475, 482, 503). For IN, we can perform the same analysis, and the values are: (476, 482, 495). These values, especially the average, are so similar that nodes of IN appear to be quite close to SCC.

Likewise, for SCC the (min, average, max) depths for termination under out-links are (430, 434, 444). For OUT, the values are (430, 434, 444).

Now, consider the probability that an ordered pair (u, v) has a path from u to v. By noting that the average in-size of nodes in IN is very small (171) and likewise the average out-size of nodes in OUT is very small (3093), the pair has a path with non-negligible probability if and only if u is in SCC + IN, and v is in SCC + OUT. The probability of this event for node pairs drawn uniformly from our crawl is only 24%; for node pairs drawn from the weak component it is only 28%. This leads to the somewhat surprising conclusion that, given a random start and finish page on the Web, one can get from the start page to the finish page by traversing links barely a quarter of the time.

The structure that is now unfolding tells us that it is relatively insensitive to the particular large crawl we use. For instance, if AltaVista's crawler fails to include some links whose inclusion would add one of the tendrils to the SCC, we know that the resulting change in the sizes of SCC and TENDRIL will be small (since any individual tendril is small). Likewise, our experiments in which we found that large components survived the deletion of nodes of large in-degree show that the connectivity of the Web is resilient to the removal of significant portions.

3.5. Diameter and average connected distance

As we discussed above, the directed diameter of the SCC is at least 28. Likewise, the maximum finite shortest path length is at least 503, but is probably substantially more than this: unless a short tube connects the most distant page of IN to the most distant page of OUT without passing through the SCC, the maximum finite shortest path length is likely to be close to $475 + 430 = 905$.

We can estimate the average connected distance using our 570 BFS start points, under both in-links and out-links. The values are shown below; the column headed 'Undirected' corresponds to the average undirected distance.

Edge type	In-links (directed)	Out-links (directed)	Undirected
Average connected distance	16.12	16.18	6.83

These results are in interesting contrast to those of [4], who predict an average distance of 19 for the Web based on their crawl of the nd.edu site; it is unclear whether their calculations consider directed or undirected distances. Our results on the other hand show that over 75% of time there is no directed path from a random start node to a random finish node; when there *is* a path, the figure is roughly 16. However, if links can be traversed in either direction, the distance between random pairs of nodes can be much smaller, around 7, on average.

4. Further work

Further work can be divided into three broad classes.

(1) More experiments aimed at exposing further details of the structures of *SCC*, *IN*, *OUT*, and the *TENDRILS*. Would this basic structure, and the relative fractions of the components, remain stable over time?

(2) Mathematical models for evolving graphs, motivated in part by the structure of the Web; in addition, one may consider the applicability of such models to other large directed graphs such as the phone-call graph, purchase/transaction graphs, etc. [3].
(3) What notions of connectivity (besides weak and strong) might be appropriate for the Web graph? For instance, what is the structure of the undirected graph induced by the co-citation relation or by bibliographic coupling [31].

Acknowledgements

We thank Keith Randall for his insights into our SCC algorithm and implementation.

References

[1] L. Adamic and B. Huberman, The nature of markets on the World Wide Web, Xerox PARC Technical Report, 1999.
[2] L. Adamic and B. Huberman, Scaling behavior on the World Wide Web, Technical comment on [5].
[3] W. Aiello, F. Chung and L. Lu, A random graph model for massive graphs, ACM Symposium on the Theory and Computing, 2000.
[4] R. Albert, H. Jeong and A.-L. Barabasi, Diameter of the World Wide Web, Nature 401 (1999) 130–131.
[5] A. Barabasi and R. Albert, Emergence of scaling in random networks, Science 286 (509) (1999).
[6] P. Barford, A. Bestavros, A. Bradley and M.E. Crovella, Changes in Web client access patterns: characteristics and caching implications, World Wide Web, Special Issue on Characterization and Performance Evaluation, 2 (1999) 15–28.
[7] K. Bharat, A. Broder, M. Henzinger, P. Kumar and S. Venkatasubramanian, The connectivity server: fast access to linkage information on the web, in: Proc. 7th WWW, 1998.
[8] K. Bharat and M. Henzinger, Improved algorithms for topic distillation in hyperlinked environments, in: Proc. 21st SIGIR, 1998.
[9] S. Brin and L. Page, The anatomy of a large scale hypertextual web search engine, in: Proc. 7th WWW, 1998.
[10] R.A. Butafogo and B. Schneiderman, Identifying aggregates in hypertext structures, in: Proc. 3rd ACM Conference on Hypertext, 1991.
[11] J. Carriere and R. Kazman, WebQuery: searching and visualizing the Web through connectivity, in: Proc. 6th WWW, 1997.
[12] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan and S. Rajagopalan, Automatic resource compilation by analyzing hyperlink structure and associated text, in: Proc. 7th WWW, 1998.
[13] S. Chakrabarti, B. Dom, D. Gibson, S. Ravi Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins, Experiments in topic distillation, in: Proc. ACM SIGIR Workshop on Hypertext Information Retrieval on the Web, 1998.
[14] S. Chakrabarti, D. Gibson and K. McCurley, Surfing the Web backwards, in: Proc. 8th WWW, 1999.
[15] J. Cho, H. Garcia-Molina, Synchronizing a database to improve freshness, To appear in 2000 ACM International Conference on Management of Data (SIGMOD), May 2000.
[16] M. Faloutsos, P. Faloutsos and C. Faloutsos, On power law relationships of the internet topology, ACM SIGCOMM, 1999.
[17] S. Glassman, A caching relay for the world wide web, in: Proc. 1st WWW, 1994.
[18] F. Harary, Graph Theory, Addison-Wesley, Reading, MA, 1975.
[19] B. Huberman, P. Pirolli, J. Pitkow and R. Lukose, Strong regularities in World Wide Web surfing, Science 280 (1998) 95–97.
[20] J. Kleinberg, Authoritative sources in a hyperlinked environment, in: Proc. 9th ACM–SIAM SODA, 1998.
[21] R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins, Trawling the Web for cyber communities, in: Proc. 8th WWW, April 1999.
[22] R. Kumar, P. Raghavan, S. Rajagopalan and A. Tomkins, Extracting large scale knowledge bases from the Web, in: Proc. VLDB, July 1999.
[23] R.M. Lukose and B. Huberman, Surfing as a real option, in: Proc. 1st International Conference on Information and Computation Economies, 1998.
[24] C. Martindale and A.K. Konopka, Oligonucleotide frequencies in DNA follow a Yule distribution, Computer and Chemistry 20 (1) (1996) 35–38.
[25] A. Mendelzon, G. Mihaila and T. Milo, Querying the World Wide Web, Journal of Digital Libraries 1 (1) (1997) 68–88.
[26] A. Mendelzon and P. Wood, Finding regular simple paths in graph databases, SIAM J. Comp. 24 (6) (1995) 1235–1258.
[27] V. Pareto, Cours d'économie politique, Rouge, Lausanne et Paris, 1897.
[28] P. Pirolli, J. Pitkow and R. Rao, Silk from a sow's ear: extracting usable structures from the Web, in: Proc. ACM SIGCHI, 1996.
[29] J. Pitkow and P. Pirolli, Life, death, and lawfulness on the electronic frontier, in: Proc. ACM SIGCHI, 1997.
[30] H.A. Simon, On a class of stew distribution functions, Biometrika 42 (1955) 425–440.
[31] H.D. White and K.W. McCain, Bibliometrics, in: Annual Review of Information Science and Technology, Vol. 24, Elsevier, Amsterdam, 1989, pp. 119–186.
[32] G.U. Yule, Statistical Study of Literary Vocabulary, Cambridge University Press, New York, 1944.
[33] G.K. Zipf, Human Behavior and the Principle of Least Effort, Addison-Wesley, Reading, MA, 1949.