Proceedings of the International Conference on Information Technology
(ITCC 2004), Las Vegas, April 5-7, 2004.

From Paragraph Networks to Document Networks

Jinghao Miao and Daniel Berleant
Dept. of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011, USA
{jhmiao, berleant}@iastate.edu

Abstract

We investigate hypertext repositories with automatically generated hyperlinks among paragraphs. The algorithms that generate the links yield results with significant common characteristics despite large differences among the algorithms. Furthermore, different repositories, generated by the same algorithms show significant common characteristics, although generated from the return lists of different search engines. Finally, repositories generated for different domains also have common characteristics. This suggests that these characteristics are pervasive properties of judiciously retrieved document sets.

The common characteristics revolve around a tendency for some documents (the kernel) to be destinations of a disproportionately high fraction of the hyperlinks in the repository. An emergent property of such repositories with practical significance is that browsing activities will have a tendency to trap the user within the kernel, perhaps without their realizing it. To enable users to choose to avoid trapping, we propose a ring structure for hypertext repositories which is exposed to the user by annotations to the hyperlinks.

Keywords:
MultiBrowser, search engines, clan graphs, n-grams, information foraging.

1. Introduction

The quantity of information on the World Wide Web (WWW) has motivated improvements in information access, manipulation, and presentation techniques to facilitate effective use of these resources. The concept of *information foraging* has been proposed as a paradigm for navigation different from the more goal-oriented activity of the traditional information retrieval paradigm ([1][9][7]). It has been suggested that this concept can help in understanding how to create new interactive information system designs ([7][14]).

To address this goal, we have been exploring information foraging within the set of documents whose URLs are provided by a Web search engine in response to a query [4]. For a search engine return list, each paragraph in each document is processed, and the 5 other paragraphs in the repository that are most similar to it are found. The distance metric we used for comparing paragraphs is the cosine measure in the space of strings of 5 characters, or 5-grams ([5][13][17]). This approach has been shown to produce results competitive with expressing texts as weighted vectors of words ([19]).

In this technique, each paragraph is expressed as a vector of 5-grams, with each 5-gram weighted by the number of times it occurs in the paragraph. Each paragraph is then mapped to a normalized point in 5-gram space ([5][13][17]), and points are compared using the cosine similarity measure. For each paragraph, the 5 other most similar paragraphs are identified regardless of what documents they are in. Based on these similarity computations, different algorithms for inserting hyperlinks among paragraphs can be used, and their properties compared. That is the subject of this report.

For each paragraph, 5 hyperlinks are added, from it to each of those similar paragraphs. The intent is to support simultaneous display of multiple related paragraphs in separate windows ([3][8][16]). This is consistent with a long-running current of thought in the hypertext field typified by such early and influential works as [10][11][14][15][18]. Therefore the entire repository contains a network of paragraphs connected by these hyperlinks. The properties of these link networks form the present topic of investigation.

Next we give some background regarding our analysis of these networks, followed by a description of the linking algorithms in Section 3. Analysis and results of the properties of the link networks for 10 separate repositories are given in Sections 4 and 5. A system design technique consistent with the results is provided in Section 6, and conclusions appear in Section 7.

2. Background

We distinguish between graphs of links among paragraphs (p-graphs) and graphs of links among documents (d-graphs). P-graphs are generated first from the repositories, and d-graphs from the p-graphs.

2.1 P-Graphs

A P-graph describes the hyperlinks among paragraphs in a repository. It is a directed graph in which each node represents a paragraph. In this investigation, paragraph *i* in a particular document has hyperlinks to 6 paragraphs, itself and the 5 others computed as most similar to it. A part of the P-graph containing node *i* is illustrated in Figure 1. The directed links indicate hyperlinks from paragraph *i* to the most similar paragraphs in the repository regardless of what documents these paragraphs are in. In the system that creates the repositories, these are implemented as a single multi-target hyperlink; clicking it brings up a six-window display that simultaneously shows the six targets [2].

2.2 D-Graphs

A D-graph is a directed graph showing which documents contain paragraph link(s) to paragraph(s) in which other documents. In a D-graph, each node represents a document in the repository. Directed links between nodes have weights calculated based on the number of paragraph links between the documents, as follows.

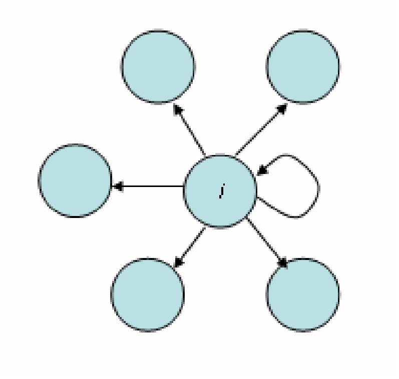

Figure 1. A paragraph contains links to the five most similar other paragraphs.

Let *A* and *B* be two documents, and let *i* and *j* be paragraphs, *i* in A and *j* in B. If there is a directed link from *i* to *j* in the P-graph, we construct a directed link from *A* to *B* in the D-graph. The weight of the *A*-*B* link is the total number of directed links in the P-graph from paragraphs in *A* to paragraphs in *B*.

Figures 2 and 3 give an example. Suppose we have a P-graph where document A has 4 paragraphs and document B has 3, and the links between paragraphs in both documents are as in Figure 2. Then the corresponding D-graph is shown in Figure 3. A *link weight* is calculated for each link in the D-graph by adding up all of the links in the P-graph that connects a paragraph in *A* to a paragraph in *B*. A *document weight* is then defined as the sum of the link weights of its incoming links. For example, document *A* has weight 2, while B has weight 5. The bigger a link weight, the more related to each other, in some sense, the two linked documents are. The bigger a document weight, the more central, in some sense, the document is within the repository. Thus the number of paragraph links in the P-graph supporting a given document link in the D-graph is encoded in the weight of the document link, while paragraph links are unweighted.

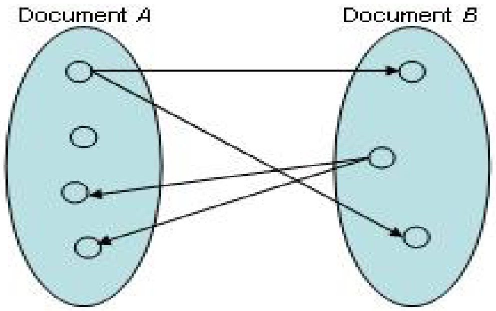

Figure 2. P-graph for documents *A* and *B*.

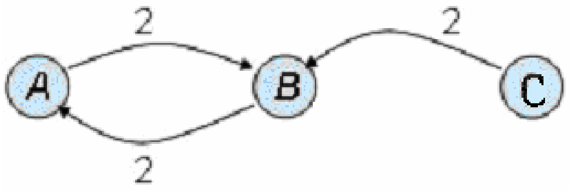

Figure 3. D-graph for documents *A* and *B* (and *C*).

The P-graphs and D-graphs both represent opportunities for navigation in the repository. A link in a P-graph represents a real hyperlink associated with a paragraph displayed in a web browser, for example. The hyperlinks associated with a paragraph could each provide a separate navigation option or, depending on system design and implementation, a single click on the paragraph could bring up a display of the different destination paragraphs in separate windows. Assuming an entire document is accessible from any paragraph in it (e.g. by scrolling the window containing the paragraph to see other parts of the document, then a link in a D-graph is followed when any inter-paragraph link connecting one document to another is followed.

2.3 Centrality and Dispersion

If some documents in the D-graph have relatively high weight and others have low or even zero weight, then a user browsing within the repository will have a tendency, perhaps without even realizing it, to move into a “core” of high-weight documents. Low-weight documents on the other hand will tend to be visited rarely or not at all. We call this phenomenon *trapping*. If a repository has a strong tendency toward trapping it is considered to have high *centrality*. However, if documents tend to have similar weights, the repository is considered to have high *dispersion*.

Our goal is to support effective navigation. This suggests avoiding trapping, while at the same time enabling the user to move to documents that are central to a repository. A good D-graph is one that has sufficient centrality to draw a user toward the core, most "relevant" documents, and sufficient dispersion to prevent other documents from being inaccessible to a greater or lesser degree.

3. P-graph Generating Algorithms

Motivated by the need to generate D-graphs with a desirable balance between centrality and dispersion, we investigated three P-graph generation algorithms to use as input to the D-graph generation process described earlier. These P-graph algorithms are described in this section. The next section investigates the D-graphs that are implied by these P-graphs.

3.1 Basic P-Graph Algorithm

This is the 5-gram base algorithm described in Section 2. The other two P-graph algorithms are based on further processing of the Basic P-graphs produced by the Basic P-graph algorithm.

3.2 2-Clan P-Graph Algorithm

An *N*-clan [20][15] is a graph in which each node has a path of up to length *N* to every other node, and all these paths contain only nodes in the clan. Such clans can appear as subgraphs of a larger graph. Previous work [20] has shown that clan graphs can help in constructing a collection of high quality. We are interested in 2-clan graphs here. Why 2-clan graphs in particular? Figure 4 graphically depicts four types of indirect inter-paragraph relationships that can be derived from the Basic P-graph. Figure 4(*a*) is simply the relationship stated in the Basic P-graph. Figure 4(*b*) shows a Basic p-graph in which paragraph *A* is related to paragraphs *B* and *C*. This suggests that *B* and *C* are related to each other by virtue of being related to *A*. Figure 4(*c*) shows a Basic P-graph in which paragraphs *B* and *C* are related to *A*. This suggests that that *B* and *C* are also related to each other by virtue of being similar to *A*. Figure 4(*d*) shows a limited (two-link) transitivity relationship in a Basic P-graph that suggests paragraph *C* is related to paragraph *B* because *C* is related to *A* and *A* is related to *B*. The links shown in Figure 4, augmented with the new relatedness links, are shown in Figure 5. *By replacing the structures in Figure 4, wherever they appear in a Basic P-graph, with the corresponding structures in Figure 5 a 2-Clan P-graph is formed.*

The relationships in Figure 5 are motivated by the 2-clan graph concept. Other relationships derivable from 3-or-more-clan graphs would be more remote and their value might be presumed more questionable.

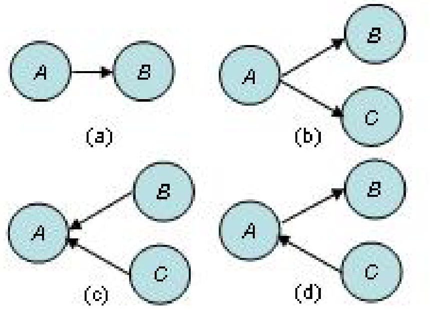

Figure 4. Four types of indirect inter-paragraph links.

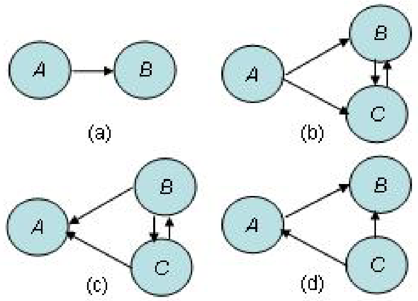

Figure 5. Situations depicted in Figure 4 with the new, derived paragraph links added.

The *2-Clan P-graph* may be used to generate a new D-graph, which will likely contain more edges than the D-graph based on the Basic P-graph because some documents are likely to be linked by paragraph connections in the 2-Clan P-graph that were not linked by paragraph connections in the Basic P-graph.

3.3 Modified 2-Clan P-Graph Algorithm

We also investigated the D-graphs implied by a p-graph formed by adding to the Basic P-graph only those links newly implied by Figure 5(*c*). The resulting p-graph will be called a Modified 2-Clan P-graph. Modified 2-Clan P-graphs tend to have more links than the corresponding Basic P-graphs, but fewer than the corresponding 2-Clan P-graph.

4. Experiments

We have now introduced three types of P-graphs: Basic P-graphs, 2-Clan P-graphs, and Modified 2-Clan P-Graphs. Each of these implies D-graphs. We investigated how these D-graphs vary across these experimental conditions: P-graph type (Experiment 1); document retrieval system (i.e. web search engine, Experiment 2); and topic (Experiment 3). Experiment 1 also provides a comparison with the well-known HITS algorithm [12], suggesting that the 2-Clan P-graph approach can provide similar results but at lower computational cost.

Experiment 1. This experiment investigated how different P-graph algorithms influence centrality and dispersion in the D-graphs they imply. The document set used was that returned by a query *powered parachuting*, submitted to the Altavista search engine. The Basic P-graph, 2-Clan P-graph and Modified 2-Clan P-graph algorithms were applied and corresponding D-graphs were generated. Based on the resulting D-graphs, documents were ranked according to their weights. Figure 6 shows the document weights implied by Basic P-graphs. Figures 7 and 8 show the document weights implied by 2-Clan P-graphs and Modified 2-Clan P-graphs respectively.

Figures 6-8 all have curves that decay in a quasi-exponential manner. This indicates relatively high centrality, implying a tendency for users to perhaps unwittingly be guided by the repository structure toward browsing documents with high weights, and away from browsing documents with low weights.

A well-known ranking algorithm, HITS, was also implemented and used to order the documents based on the Basic P-graph. The purpose was to compare its results to the Basic, 2-Clan, and Modified 2-Clan P-graph algorithms. Figure 9 shows the histogram of the document weights implied by the 2-Clan P-graph method. Figure 10 shows the Hub and Authority scores of the documents implied by the HITS algorithm. In both graphs the same repository as in Figures 6-9 was used, but only half of the documents, those with the highest weights, are shown.

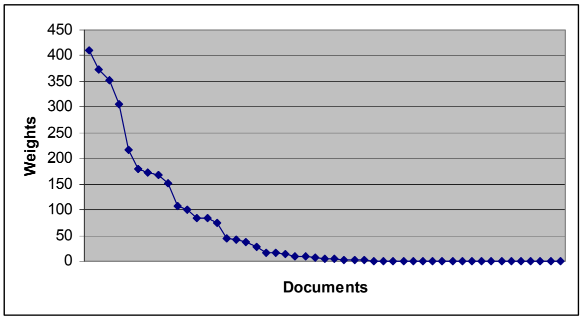

Figure 6. Weights of documents in the D-graph of the powered parachuting repository resulting from the Basic P-graph.

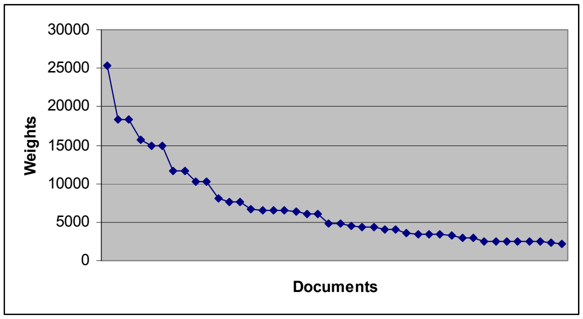

Figure 7. Weights of documents in the D-graph of the powered parachuting repository resulting from the 2-Clan P-graph.

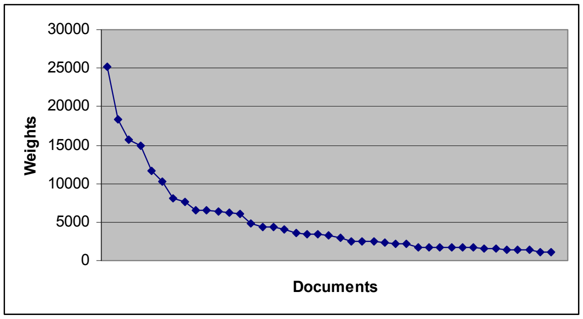

Figure 8. Weights of documents in the D-graph of the powered parachuting repository resulting from the Modified 2-Clan P-graph.

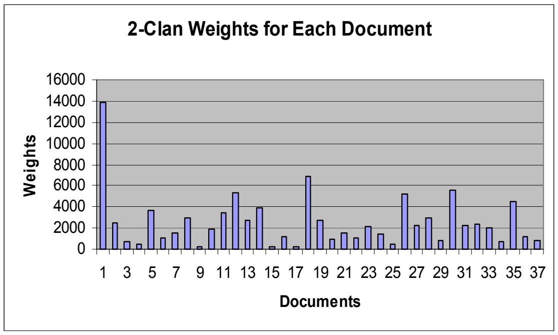

Figure 9. Document weights implied by the 2-Clan P-graph generated for a repository of documents returned by Altavista.

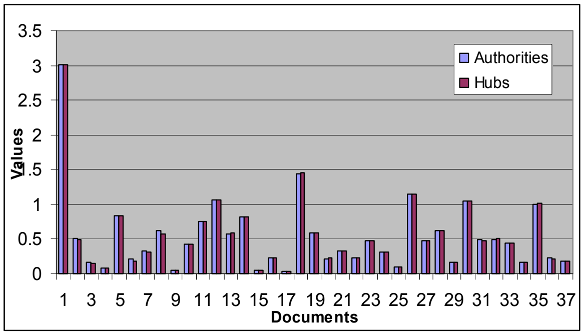

Figure 10. Authority and Hub values for each document in the same document set as in Figure 9. Values were generated using HITS on the Basic P-graph.

A comparison of Figures 9 and 10 shows that the relative weights of the documents in the D-graph implied by the 2-Clan P-graph are very similar to the relative document weights implied by the HITS algorithm as applied to the Basic P-graph. Yet, the 2-Clan P-graph algorithm is computationally more efficient because it generates document weights in one pass, while the HITS algorithm is iterative.

Experiment 2. This experiment compared different search engines in terms of centrality and dispersion in D-graphs generated from their returned document lists. We selected five popular search engines, Google, Altavista, Hotbot, Lycos and Ask Jeeves, to generate repositories based on the query *vegetarian cooking*? As in the previous experiment, we applied the Basic P-graph, 2-Clan P-graph and Modified 2-Clan P-graph algorithms to generate P-graphs and their implied D-graphs, and then computed the weights of the documents from each D-graph.

Figures 11 through 13 show the results. These figures show that the search engines all return document lists with similar characteristics. In particular, they lead to D-graphs whose document weights, when ordered high-to-low, give curves with high centrality and thus low dispersion.

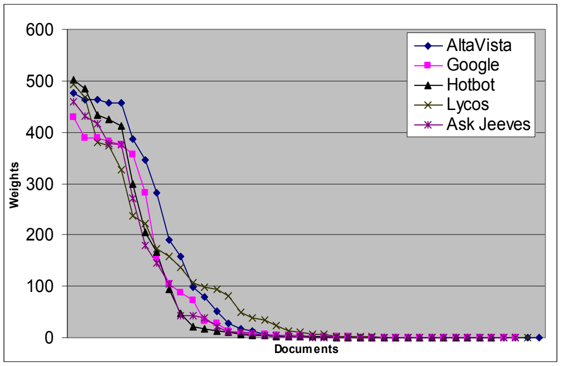

Figure 11. Weights of documents, ordered high-to-low, from the D-graphs of each of five vegetarian cooking repositories derived from the return lists of five Web search engines. The D-graphs were derived from the Basic P-graphs of the repositories.

Experiment 3. This experiment examined if quasi-exponential curves (and therefore high centrality and low disperson) occur across different topics. Repositories were generated based on documents returned in response to two different queries, *powered parachuting* and *vegetarian cooking*, to five search engines. The two queries had widely varying total numbers of relevant documents returned, one almost 100 times the other. Thus these repositories sample the diversity of repositories in both topic and total number of relevant documents. Figures 14 through 16 show the results from the *powered parachuting* query using different paragraph linking algorithms. A comparison with Figures 11 through 13 shows that both queries resulted in repositories with the same quasi-exponential centrality and dispersion properties.

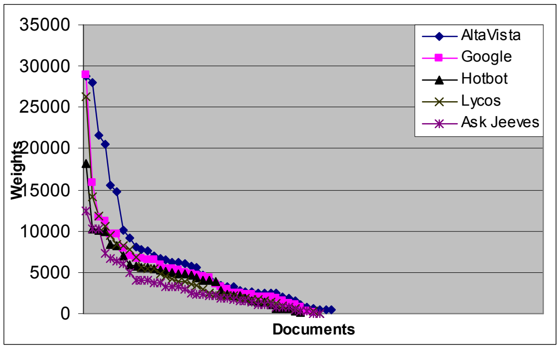

Figure 12. Weights of documents, ordered high-to-low, from the D-graphs of each of five vegetarian cooking repositories derived from the return lists of five Web search engines. The D-graphs were derived from the 2-Clan P-graphs of the repositories.

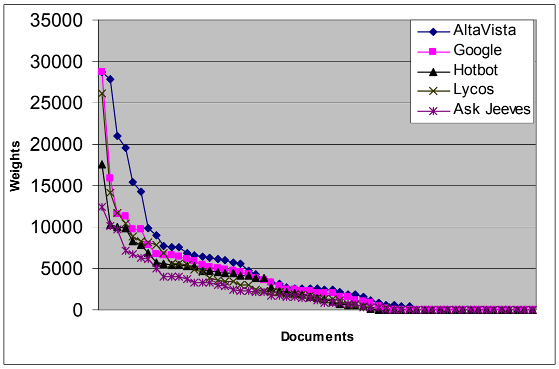

Figure 13. Weights of documents, ordered high-to-low, from the D-graphs of each of five vegetarian cooking repositories derived from the return lists of five Web search engines. The D-graphs were derived from the Modified 2-Clan P-graphs of the repositories.

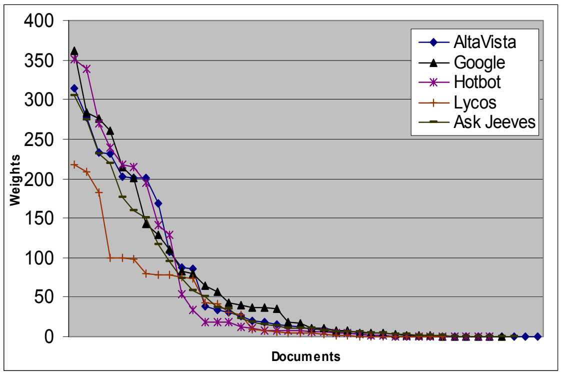

Figure 14. Weights of documents, ordered high-to-low, from the D-graphs of each of five powered parachuting repositories derived from the return lists of five Web search engines. The D-graphs were derived from the Basic P-graphs of the repositories.

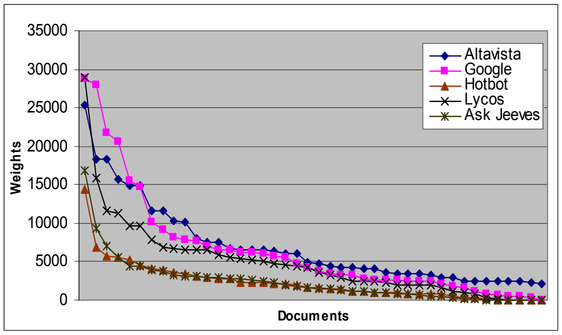

Figure 15. Weights of documents, ordered high-to-low, from the D-graphs of each of five powered parachuting repositories derived from the return lists of five Web search engines. The D-graphs were derived from the 2-Clan P-graphs of the repositories.

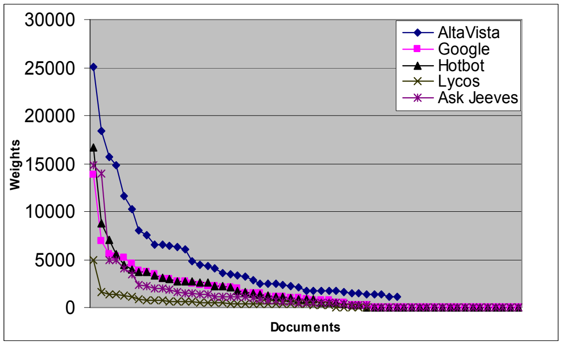

Figure 16. Weights of documents, ordered high-to-low, from the D-graphs of each of five powered parachuting repositories derived from the return lists of five Web search engines. The D-graphs were derived from the Modified 2-Clan P-graphs of the repositories.

5. Results

The three experiments indicate a tendency toward quasi-exponential curves that occurs across varied conditions. In particular, it occurred across two contrasting document sets, across multiple search engines and across different p-graph algorithms.

The quasi-exponential curve shapes suggest a Zipf-like relationship [21][22][23]. This relationship describes the empirical observation of constant slope for a log-log plot of rank vs. value of ranked parameter. In this case the rank is obtained by ordering documents from highest to lowest weight, and the value of the ranked parameter is therefore document weight. To see how well the Zipf relationship applies here, we took the 30 curves in Figures 12 through 16 and normalized each curve to have the same height. Then the average height for each x-axis value was computed over all 30 curves. Finally the resulting curve was plotted on a log-log scale. Figure 17 shows the result. Although usually applied to larger data sets, the results here show a curve of fairly constant slope over most of its span, but with a noticeable drop-off toward the end where the graininess of the data is largest since document weights are low (a zero weight could not even be plotted on a log scale). Because of the general pervasiveness of Zipf-type curves in text, (1) it is not surprising that it appears here, and (2) it is likely that it applies in other hypertext scenarios as well. This underlines the need to watch for high centrality and consequently trapping in hypertext repositories, and to enable users to avoid trapping when they wish to.

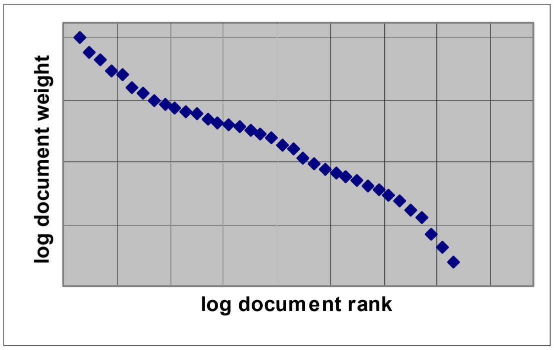

Figure 17. Log-log plot of document weight vs. document rank.

The high degree of centrality in the relevant figures suggests that in each case, a kernel of documents that are most similar has been identified. They presumably reflect the focus of the repository, with other documents being both figuratively and literally in the tail of the curve. However from an information foraging point of view, high centrality implies that users, as they follow links, are likely to move toward and into those documents with the highest weights, and to do so without necessarily realizing it [6][7]. This is potentially a problem because they will fail to be exposed to the rich variety available in the repository.

Under the information foraging paradigm, this is a problem. It is exacerbated by the very low; sometimes even zero weights of many of the documents, which tends to make them inaccessible, again preventing the user from experiencing the richness of the repository. The next section proposes a solution, currently implemented in our MultiBrowser system [2].

6. Navigation in Ring-Structured Repositories

We propose a ring-structured solution to support an information foraging style of navigation in repositories with high centrality. The ring structure consists of a kernel containing the documents with highest weights, and two concentric rings around it containing documents of intermediate and low weights, respectively (Figure 18).

During the process of navigation, the user should be presented with links that are annotated to indicate whether they point to documents in the kernel, near ring or far ring. The user is thus able to decide which links to follow when foraging. The kernel intuitively would be associated with documents of high relevance to whatever criteria or query was used to obtain documents for the repository. The near ring intuitively would contain documents that are likely to be of interest if foraging beyond the kernel is desired. The far ring would contain documents peripherally related to the focus of the repository and thus most likely to be of interest when foraging goals are more expansive or undirected.

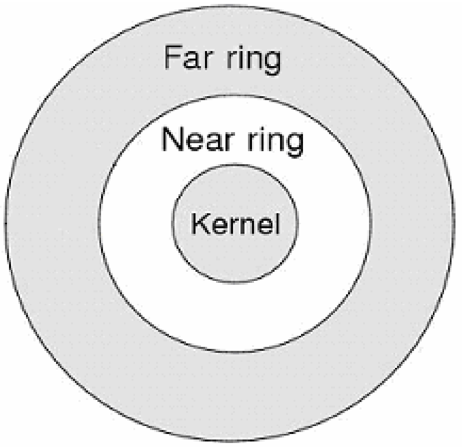

Figure 18. Ring structuring of a hypertext repository classifies documents into the kernel, near ring, or far ring.

Suppose each paragraph in a repository has six outgoing links, 5 to other paragraphs and 1 to itself. These links might, as in the MultiBrowser system [2], be implemented as a 6-target hyperlink that, when clicked, brings up a 6-window display of all six targets. Let k be the number of targets in kernel documents. Likewise, let n and f be the numbers of targets in paragraphs contained in the near and far ring documents respectively. Then we have the following equation:

$$k + n + f = 6.$$

Clicking the 6-target hyperlink brings up a 6-window display showing k paragraphs in kernel documents, n paragraph in near-ring documents, and f in far-ring documents. We introduce a parameter r to describe the degree to which the 6-target hyperlink emanating from a paragraph tends to direct browsing toward the kernel:

$$r = \frac{100 - 8 \times n - 16 \times f}{100}.$$

If $n = 0$ and $f = 0$ that means all targets are paragraphs contained in kernel documents, and $r = 1$. If $n = 6$ and $f = 0$ that means all targets are paragraphs contained in near ring documents and r is close to 0.50, indicating that the tendency of the 6-target hyperlink paragraph to direct browsing toward the kernel is intermediate. If $n = 0$ and $f = 6$, all targets are paragraphs contained in far-ring documents and r is close to 0, indicating a 6-target hyperlink that points away from the kernel. The trapping problem is alleviated because hyperlinks are annotated with numbers that give the guidance needed for a user to forage without being trapped in a small subset of the documents.

7. Conclusions

Quasi-exponential Zipf-type curves characterize automatically generated paragraph-linked hypertext repositories under varied conditions. These curves depict document weights showing that some documents in a repository of query-retrieved documents are the destinations of many more links than other documents, and other documents are the destinations of very few links. Although the full extent to which these results generalize has not been determined, their consistent appearance across varied experimental conditions suggests that the degree to which the phenomenon generalizes to hypertext in general is significant and may be pervasive. Given hypertext repositories with this quasi-exponential property, it is important to enable users to choose a foraging strategy that will allow them, for example, to avoid becoming perhaps unwittingly trapped within the repository kernel. A ring-like partitioning of the repository in which documents are categorized into the kernel, the near ring, or the far ring, as well as annotations to hyperlinks that provide numerical clues, was proposed that would provide users with the required information to inform their browsing activities.

References

[1] Berleant, D. and H. Berghel. "Customizing Information: Part 1," *Computer,* Sept. 1994, **27** (9): 96-98. "Part 2," Computer, Oct. 1994, **27** (10): 76-78.

[2] Berleant, D., J. Miao, Z. Gu and D. Xu. "Towards Dialogues With Documents: MultiBrowser," submitted. http://class.ee.iastate.edu/berleant/home/.

[3] Bly, S.A. and J. K. Rosenberg. “A Comparison of Tiled and Overlapping Windows,” Proceedings of *Proc. ACM Conf. Human Factors in Computing Systems* (*CHI ’86*), 1986, **17**: 101-106.

[4] Brin, S. and L. Page. “The Anatomy of Large-Scale Hypertextual Web Search Engine,” *Proceedings of the WWW 7 Conference*, 1998, 107-117.

[5] Damashek, M. “Gauging Similarity With N-Grams: Language-Independent Categorization of Text,” *Science*, 10 Feb. 1995, **267:** 843-848.

[6] Foltz, M.A. “Designing Navigable Information Space,” M.S Thesis, Dept. of Electrical and Computer Engineering, MIT, 1998.

[7] Furnas, G.W. “Effective View Navigation,” *Proc. ACM Conf. Human Factors in Computing Systems* (*CHI ’97*), 1997, 367-374.

[8] Halasz, F.G. “Reflections on Notecards: Seven Issues For the Next Generation of Hypermedia Systems,” *Communications of the ACM*, 1988, **31** (7): 836-852.

[9] Hara, Y. and K. Watanabe. “Hypermedia Research at C&C Research Labs, NEC USA,” *CHI’97 Electronic Publications: Organizational Overview*, 1997.

[10] Kandogan, E. and B. Shneiderman. “Elastic Windows: A Hierarchical Multi-Window World-Wide Web Browser,” *ACM Symposium on User Interface Software and Technology*, 1997.

[11] Kandogan, E. and B. Shneiderman. “Elastic Windows: Evaluation of Multi-Window Operations,” *Proc. ACM Conf. Human Factors in Computing Systems* (*CHI ’97*).

[12] Kleinberg, J. “Authoritative Sources in a Hyperlinked Environment,” *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1998, **46**: 604-632.

[13] Mayfield, J. and P. McNamee. “Indexing Using Both N-Grams and Words,” *NIST Special Publication 500-242: The Seventh Text Retrieval Conference (TREC 7)*, 1998, 419-424.

[14] Pirolli, P. and S. Card. “Information Foraging in Information Access Environments,” *Proc. ACM Conf. Human Factors in Computing Systems* (*CHI ’95*), 1995, 51-58.

[15] Scott, J. “*Social Network Analysis: A Handbook*,” Sage Publications, Inc., Thousand Oaks, CA, 1991.

[16] Shneiderman, B., C. Plaisant, R. Botafogo, D. Hopkins, and W. Weiland. “Designing to Facilitate Browsing: A Look Back at the Hyperties Workstation Browser,” *Hypermedia*, 1991, **3** (2): 101-117.

[17] Shneiderman, B., D. Byrd, and B. Croft. “Sorting Out Searching, A User-Interface Framework for Text Searches,” *Communications of the ACM*, April 1998, **41** (4): 95-98.

[18] Stotts, D. and R. Furuta. “Petri-Net-Based Hypertext: Document Structure with Browsing Semantics,” *Transactions on Information Systems*, Jan. 1989, **7** (1): 3-29.

[19] Tauscher, L. and S. Greenberg. “Revisitation Patterns in World Wide Web Navigation,” *Proc. ACM Conf. Human Factors in Computing Systems* (*CHI ’97*), 399-406.

[20] Terveen, L. and W. Hill. “Constructing, Organizing, and Visualizing Collections of Topically Related Web Resources,” *ACM Trans. on CHI*, March 1999, **6** (1): 67-94.

[21] Zipf, G.K. *The Psychobiology of Language*, 1935, Houghton Mifflin.

[22] Zipf, G.K. “The Meaning-Frequency Relationship of Words,” *The Journal of General Psychology*, 1945, **33**: 251-256.

[23] Zipf, G.K. “The repetition of words, time-perspective, and semantic balance,” *The Journal of General Psychology*, 1945, **32**: 127-148.