A GRAPH BASED APPROACH TO COMPARING SEARCH ENGINES

© Jinghao Miao
Dept. of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011, USA
jhmiao@iastate.edu

© Daniel Berleant
Dept. of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011, USA
berleant@iastate.edu

Abstract

In the investigation of paragraph-linked hypertext repositories, the algorithms generating link networks have in-depth impacts on the phenomena of centrality or dispersion that are identified as important issues to be dealt with in the process of information foraging. Furthermore different repositories, generated through different search engines, even produced by the same algorithms, still shown significant variations with different degrees of centrality or dispersion. We have studied this using the MultiBrowser system as a testbed, and applying some algorithms to repositories consisting of documents downloaded based on five different search engine return lists. We study the hyperlinks added automatically to these repositories, which make these documents navigable from other documents within the repositories. The property of centrality and dispersions of linked repository can be further investigated.

Keywords:
MultiBrowser, search engines, direct display, multidisplay, clustering, Clan Graph, N-Grams, information foraging.

1. Introduction

Information flooding over the World Wide Web (WWW) has motivated researchers and developers to greatly improve intelligent information access, manipulation, and presentation techniques facilitate use of these resources, and meanwhile enabled the information foraging theory being proposed to achieve information navigation optimality by retaining relevance in presented information as compared to traditional information retrieval. It is believed that this theory can help in gaining and making sense out of information, and understanding how to create new interactive information system designs [8].

To address the same goal, we have been exploring information foraging with the MultiBrowser system [2], designed with the objective of supporting interaction with a repository of documents of approximately book length. The MultiBrowser system then processes each paragraph in each document, and finds 6 most similar paragraphs in the repository. A number of "FIND SIMILAR" hyperlinks are added representing navigating from the given paragraph to its 6 most similar paragraphs. Therefore the entire repository can be viewed as a network of linked paragraphs. The properties of the link networks carry useful information for effective foraging.

2. The testbed – MultiBrowser

The MultiBrowser system supports information foraging within hypermedia repositories. It takes as input either a search engine query or an HTML file containing links to URLs, and presents information using direct multidisplay [2], which is a type of collage-based computer browsing method. Each of the sub-windows separately displays the actual contents of information rather than the presentation of meta-contents. Collectively the sub-windows provide a view of the repository.

MultiBrowser retrieves the documents in the repository and clusters them into three groups. The k-means algorithm, a standard clustering algorithm are used. The distance metric we use is the cosine measure in the space of strings of 5 characters, or 5-grams (Damashek 1995 [17]). The distances of each document to the centroids of the three clusters are used to compute intensities of red, green, and blue (RGB) components, which are combined into a color for labeling the document. Each color bar also contains a link that, when clicked, loads the corresponding document into the full browser frame for a closer look. After a color bar has been computed for each document, the documents are segmented into paragraphs and the set of paragraphs is processed. Each paragraph is mapped to a normalized point in 5-gram space, and points are compared using the cosine similarity measure mentioned earlier. For each paragraph, the six other most similar paragraphs are identified, regardless of what documents they are in, so that all six paragraphs can later be displayed simultaneously in the six sub-windows in response to a user click on a "FIND SIMILAR" link following the given paragraph. Thus each link in essence has 6 targets. Such multi-tailed

Proceedings of the 5th Russian Conference on Digital Libraries RCDL2003, St.-Petersburg, Russia, 2003.

links have been explored as early as the '80s (Stotts and Furuta 1989 [19]). Different algorithms for computing paragraph similarity can be used, and their effects compared. That is the subject of this report.

3. Preliminary knowledge

3. 1 P-Graphs

A P-graph, by definition, describes interactions between paragraphs in a repository as a directed graph where each node represents a paragraph in the repository. In the environment of a MultiBrowser repository, suppose a paragraph i in a particular document has a hyperlink to its 6 most similar paragraphs. A part of the P-Graph containing node i is illustrated as Figure 1. The directed links indicate traversing along paragraph links to find the most similar paragraphs in the repository regardless of whichever documents within the repository these paragraphs belong to.

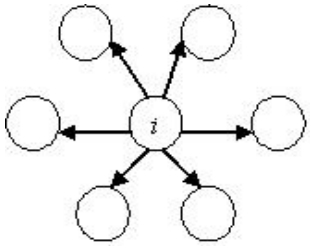

Figure 1. A paragraph contains links to the six most similar other paragraphs.

3.2 D-Graph

A D-graph, by definition, is a directed graph showing which documents contain paragraphs link(s) to paragraphs in other documents. In a D-Graph, each node represents a document in the repository. Directed links between nodes have weights, which is the number of paragraph links going between some paragraphs in one document and some paragraphs in the other, in the same direction as the D-Graph link.

Suppose A and B are two documents, and i and j are two paragraphs, one in each document respectively. If there is a directed link from i to j, we construct a directed link from A to B. The weight on the link from A to B is the total number of directed links from paragraphs in A to paragraphs in B. This is a simple example of a two-clan D-Graph. Suppose we have a P-Graph as shown in Figure 2, where document A has 4 paragraphs and document B has 3, and the links between paragraphs in both documents are represented. Then the corresponding D-Graph is as shown in Figure 3. For each document node in the D-Graph, a weight is calculated by adding up all the weights on the incoming links for that node. For example, documents A and B in Figure 3 have weights of 2 respectively. The bigger the weight on the link, the more relevant to each other the two documents are estimated to be.

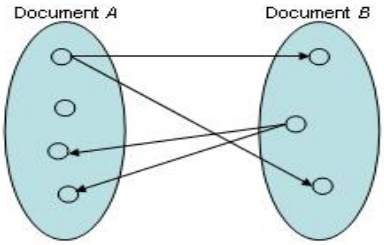

Figure 2. P-Graph of documents A and B.

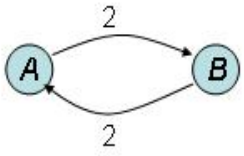

Figure 3. D-graph of documents A and B.

The P-Graphs and D-Graphs both represent opportunities for navigation in the repository. A link in a P-Graph is followed by clicking the "FIND SIMILAR" link following that paragraph (simultaneously following 5 other links as well). A link in a D-Graph is followed when an inter-paragraph link connecting one document to another is followed, which may require scrolling within a frame to get to a paragraph with such a link. Note that the number of nodes in the D-Graph is less than the number in the P-graph because there are fewer documents than paragraphs, generally, and the number of links in the D-graph is generally less than the number in the P-graph because there is at most one link from one document to another, regardless of how many paragraph links imply it. However 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.

3.3 Centrality and Dispersion

If some documents in the D-Graph have many incoming links indicating a high relevance in the repository, and others have few or even none indicating a low relevancy, then as a user navigates in the repository, this browsing process will have a tendency to move the user to a core of document that have many incoming links, while documents with few or none incoming links will tend to be visited only rarely. This "downhill" movement can result in some documents being or becoming inaccessible, if they have no incoming links. This is called trapping. If a repository has a strong downhill character, it is considered highly centralized. On the other hand, if there is little downhill tendency, it is considered dispersed.

4. D-graphs generations for analysis

In this section, we review how the P-Graph of a repository is created, and then how it is analyzed using the D-Graph concept.

4.1 Generating the P-Graph Using 5-Gram

As stated in Section 2, each document in a repository is segmented into paragraphs. The set of paragraphs are then processed by converting each into a vector in a 5-gram space. Then we apply the cosine similarity metric in the vector space to identify the 6 most similar other paragraphs for each paragraph. In practical navigation, a user is allowed to travel across the directed links from a given paragraph to the other six most similar paragraphs by clicking the "FIND SIMILAR" links. So we simply add directed links from each paragraph to its 6 most similar documents computed using 5-gram. We wish to analyze the connectivity of the documents using this P-Graph as a foundation.

4.2 2-Clan Algorithms Based on P-Graphs

In the context of MultiBrowser, nodes of N-clan graphs stand for paragraphs and directed edges stand for links from one paragraph to another. An N-clan [4] is a graph in which each node has a path to every other node with a path length up to length N, and all these paths contain only nodes in the clan. Previous work [4] has shown that the clan graph helps in constructing a collection of high quality in terms of well structured element relationships. Now we are particularly interested in 2-clan graphs here. Why 2-clan? Figure 4 graphically depicts four types of indirect inter-paragraph similarity relationships that can be inferred in the 5-gram P-Graph. Figure 4(a) directly describes the similarity stated in 5-gram P-Graph. Figure 4(b) shows paragraph A is similar to paragraphs B and C. This suggests that B and C are similar to each other by virtue of being similar to A. A link expressing this similarity has been added in correspondence to Figure 5(b). Figure 4(c) shows a P-Graph where paragraphs B and C are similar to A. This suggests that that B and C are also similar to each other by virtue of being similar to A. A link expressing this similarity has been added in correspondence to Figure 5(c). Figure 4(d) shows limited (two-link) transitivity relationship that suggests paragraph C is similar to paragraph B because C is similar to A and A is similar to B. A link expressing this similarity has been added in correspondence to Figure 5(d). The four types of relationships may be found within 2-clan graphs. We chose the 2-clan graph concept to define paragraph similarity relationships that build upon the basic 5-gram cosine metric computation of graph similarities, because relationships analogous to those of Figure 4 but found in 3-clan graphs would often be more remote and questionable.

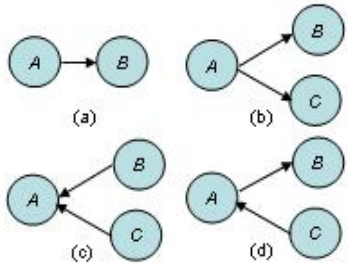

Figure 4. Four types of indirect inter-paragraph links.

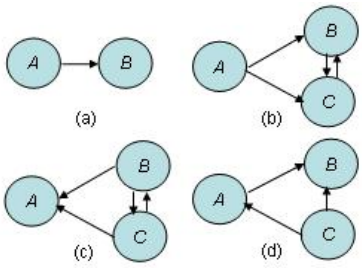

Figure 5. Situations depicted in Figure 5 with the new, derived paragraph links added.

4.3 From P-Graph to 2-Clan P-Graph

We take P-Graph generated by the 5-gram algorithm as a foundation to derive a larger set of paragraph connection graphs containing not only the original inter-paragraph connections but also the additional connections found using the derived similarity relationships of Figure 5(a-d) based on 2-clan algorithms. This set of derived connections may then be used to generate a D-Graph which will typically contain more edges than the D-Graph based on 5-gram P-Graphs because some documents are likely to be linked by paragraph connections in the derived connection set that were not linked by paragraph connections in the original set.

4.4 Simplified Two-Clan Based Algorithm

We also investigated the D-graphs implied by adding to the P-graph only those links newly implied by Figure 5(c). The results would have more links than the original 6 per paragraph, but fewer than those implied by Figure 5(b-d).

5. Analysis

So far we have introduced some algorithms to generate D-Graphs for a given repository. However, repositories containing different collections of documents may carry different properties for their D-Graphs intuitively because interactions between documents vary in terms of topics and content structures for two different repositories. Even in two repositories with similar topics, the D-Graphs may still show significant differences in their D-Graphs. To investigate the influences of repository properties on their D-Graphs, we did some experiments described as follows. We selected two topics: powered parachuting and vegetarian cooking, for generating D-Graphs from

document sets returned by five different search engines including AltaVista, Google, Hotbot, Lycos and Ask Jeeves (Table 1). Most internet surfers are familiar with the first four commercial search engines because of their powerful searching abilities and desirable searching results, but they process keyword query typically. Ask Jeeves is able to process question query, that is, a user can input a question and the search engine will return the related URLs to answer this specific question. So Ask Jeeves is a leading provider for natural language and question answering technologies.

Table 1. Comparisons of Query Types for Five Search Engines.

Google, AltaVista, Lycos, Hotbot	Keyword Based
Ask Jeeves	Keyword & Question Based

Different linking algorithms as described in Section 5 are applied to generate three different P-Graphs: 5-gram based P-Graphs with 6 outgoing links from each paragraph, 2-clan P-Graphs containing all of the new links implied by Figure 5, and P-Graphs also containing only those new links implied by Figure 5(c). We did the experiments to see if centrality and dispersion properties occur for the D-Graphs resulting from these three P-Graph algorithms

5.1 Centrality in 5-Gram Based P-Graphs

Figure 6 and Figure 7 show the weights of documents in the D-Graph for the two repositories from the document sets returned by AltaVista using the basic 5-gram algorithm to generate the P-Graph and then the D-Graph from that. Note how centralized both the two D-Graphs are, with many documents having weights of zero and many others having very small weights, while a few have extremely high weights. As mentioned above, high weights show a large number of incoming links to a particular document. The property of centrality indicates that a user would quickly be trapped into overly centralized documents and unlikely to see many others during information foraging process.

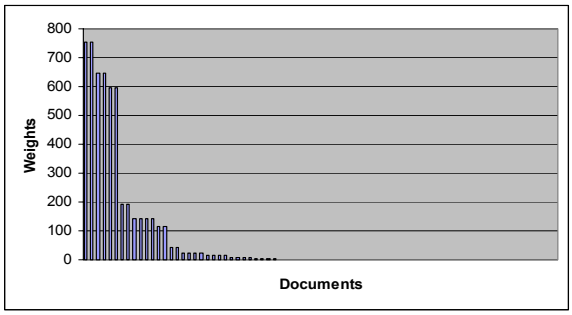

Figure 6. Weights of documents in the powered parachuting repository from document sets returned by AltaVista using 5-gram-based paragraph connections.

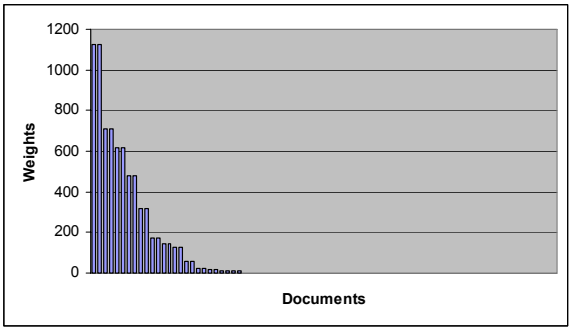

Figure 7. Weights of documents in the vegetarian cooking repository from document sets return by AltaVista using 5-gram-based paragraph connections.

The centrality in the repositories generated by basic 5-gram algorithm is no exception. To prove this point, similar experiments have been carried out on the repositories containing 40 documents for each with the same topics but generated through document sets returned by other four search engines: Google, Hotbot, Lycos and Ask Jeeves. Figure 8 through Figure 15 show the weights of documents in the D-Graphs for the two repositories of powered parachuting and vegetarian cooking using the above four search engines. The occurrence of centrality is apparent in all experiment results, which indicates a close connection between basic 5-gram algorithms and document centrality.

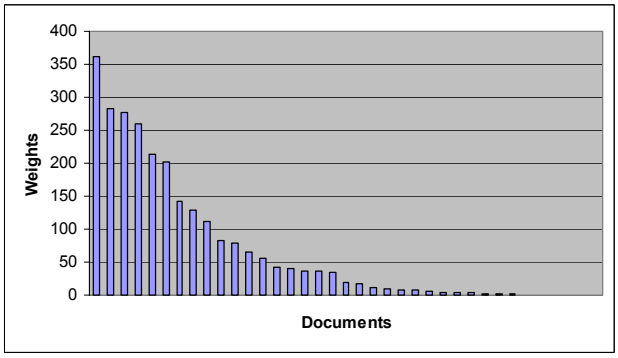

Figure 8. Weights of documents in the powered parachuting repository from document sets returned by Google using 5-gram-based paragraph connections.

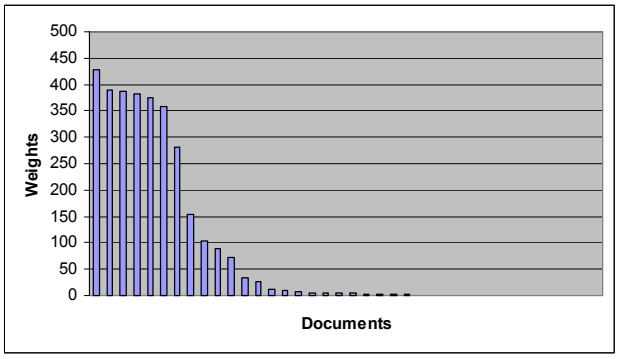

Figure 9. Weights of documents in the vegetarian cooking repository from document sets return by Google using 5-gram-based paragraph connections.

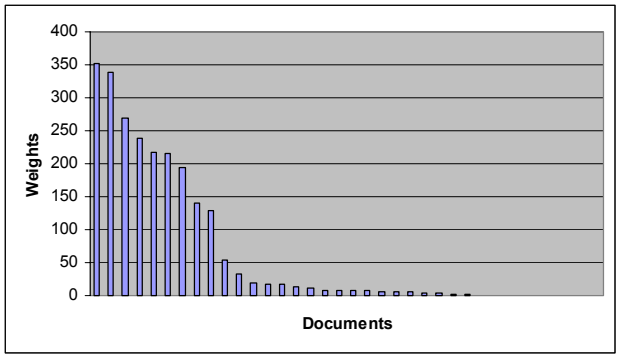

Figure 10. Weights of documents in the powered parachuting repository from document sets returned by Hotbot using 5-gram-based paragraph connections.

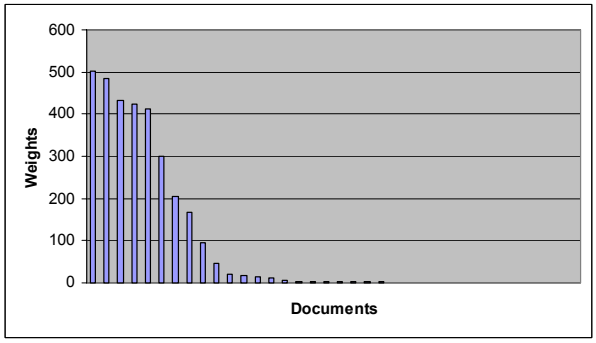

Figure 11. Weights of documents in the vegetarian cooking repository from document sets return by Hotbot using 5-gram-based paragraph connections.

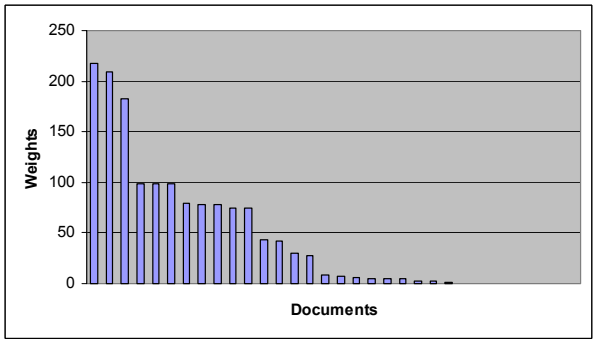

Figure 12. Weights of documents in the powered parachuting repository from document sets returned by Lycos using 5-gram-based paragraph connections.

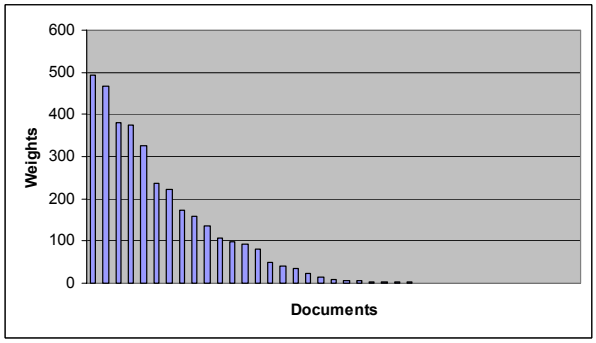

Figure 13. Weights of documents in the vegetarian cooking repository from document sets returned by Lycos using 5-gram-based paragraph connections.

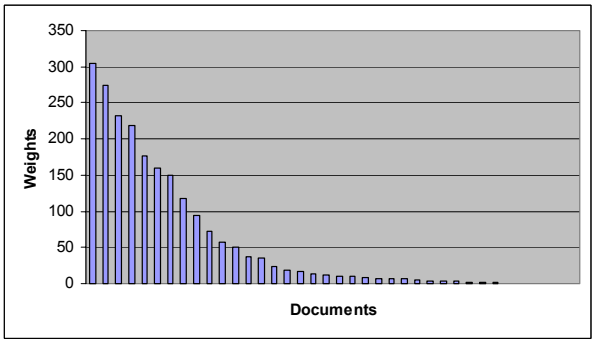

Figure 14. Weights of documents in the powered parachuting repository from document sets returned by Ask Jeeves using 5-gram-based paragraph connections.

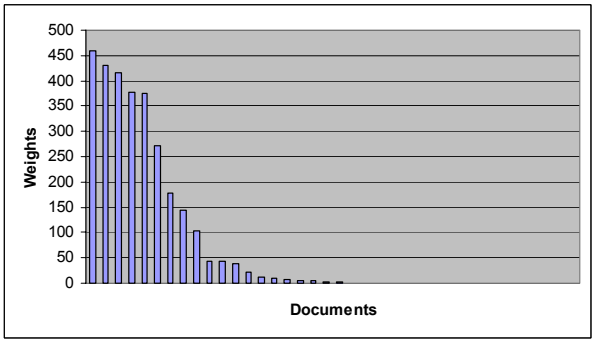

Figure 15. Weights of documents in the vegetarian cooking repository from document sets returned by Ask Jeeves using 5-gram-based paragraph connections.

Presumably the high degree of centrality in the figures indicates that a kernel of documents that are most similar has been identified. They usually reflect the focus and topic of the entire repository, with other documents being both figuratively and literally in the tail of the curve. However from an information foraging point of view, it seems undesirable to generate a repository with many inaccessible documents. This has motivated the ideas of generating D-Graphs from P-Graphs that have facilitating extra links as described in the previous section. If these D-Graphs have better properties, this would indicate the desirability of generating repositories that allow navigation based on these facilitating extra links.

5.2 Dispersion and Centrality Alleviation in Augmented 2-Clan Based P-Graphs

Because of the trapping problem in Section 5.1, we tested another graphing method, that is, 2-clan based D-Graphs. In this experiment, we first computed the weights of documents in D-Graphs constructed from 2-clan P-Graphs containing additional links on versions of the two repositories each of which has only the 22 documents with the highest ranking according to the AltaVista search engine. Figure 16 and Figure 17 show the results. Obviously the document weights are highly dispersed with insignificant variances in weights among documents. The centrality phenomenon is greatly alleviated and trapping is no longer a problem here.

However, we did not know if the dispersion phenomenon is common in repositories generated

through different search engines if 2-clan algorithm is used. Similar to what we did in testing the basic 5-gram-based P-Graphs, we did the same experiments on the other four search engines to test our hypothesis that the augmented 2-clan P-Graphs helps alleviate centrality phenomenon, but brings up high dispersion. The results are shown as Figure 18 through Figure 25.

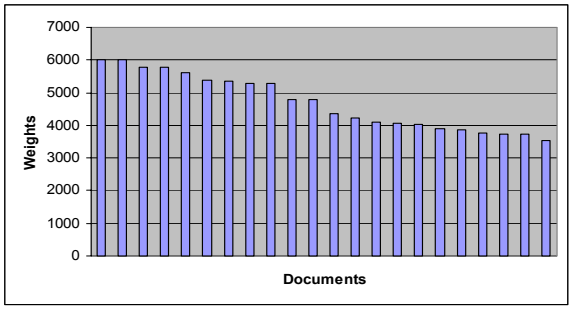

Figure 16. Weights of documents in the powered parachuting repository from document sets returned by AltaVista using 2-clan-based paragraph connections.

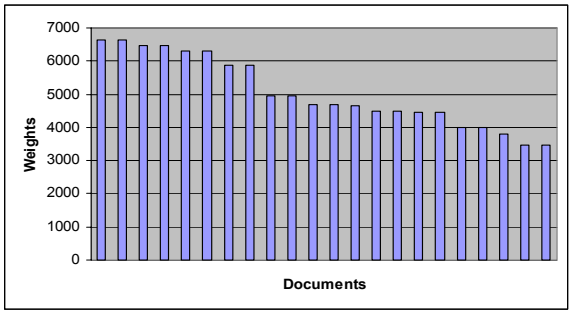

Figure 17. Weights of documents in the vegetarian cooking repository from document sets returned by AltaVista using 2-clan-based paragraph connections.

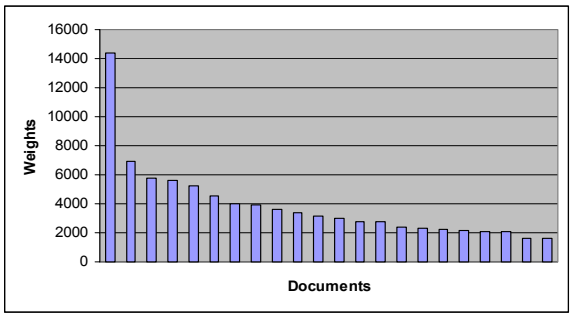

Figure 18. Weights of documents in the powered parachuting repository from document sets returned by Google using 2-clan-based paragraph connections.

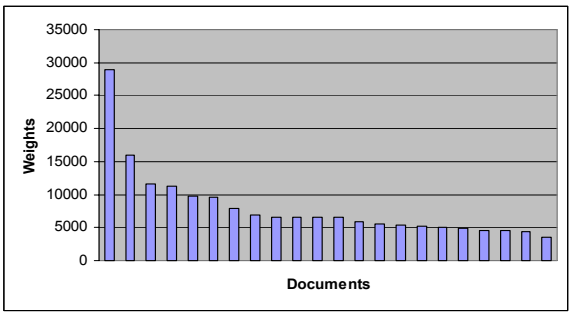

Figure 19. Weights of documents in the vegetarian cooking repository from document sets returned by Google using 2-clan-based paragraph connections.

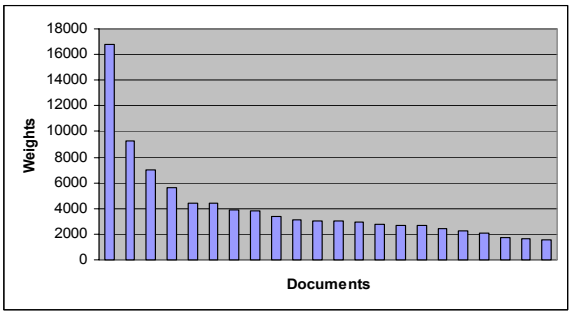

Figure 20. Weights of documents in the powered parachuting repository from document sets returned by Hotbot using 2-clan-based paragraph connections.

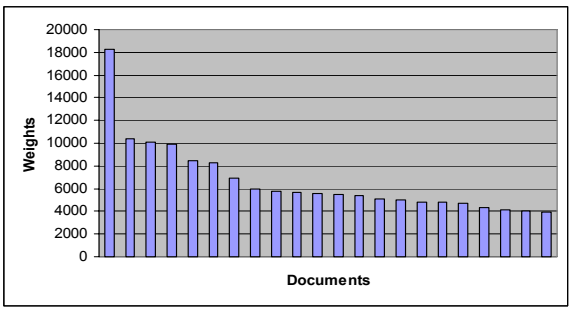

Figure 21. Weights of documents in the vegetarian cooking repository from document sets returned by Hotbot using 2-clan-based paragraph connections.

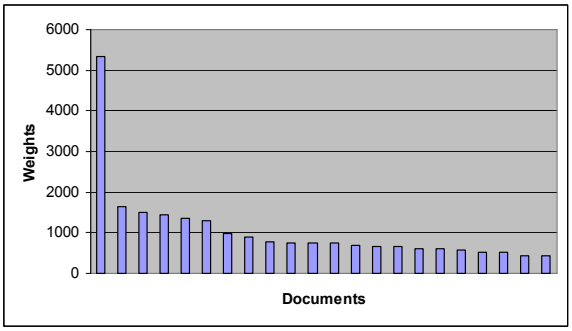

Figure 22. Weights of documents in the powered parachuting repository from document sets returned by Lycos using 2-clan-based paragraph connections.

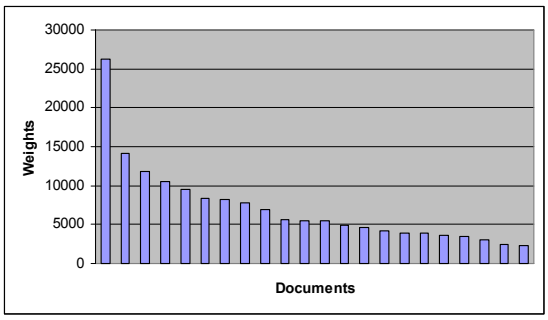

Figure 23. Weights of documents in the vegetarian cooking repository from document sets returned by Lycos using 2-clan-based paragraph connections.

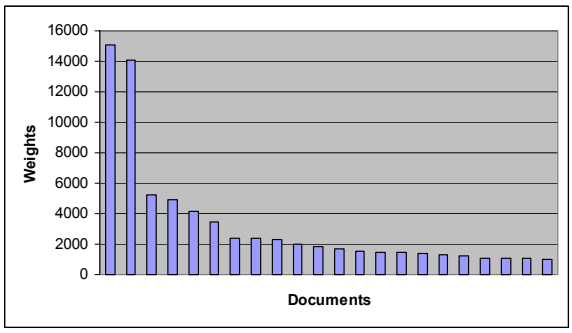

Figure 24. Weights of documents in the powered parachuting repository from document sets returned by Ask Jeeves using 2-clan-based paragraph connections.

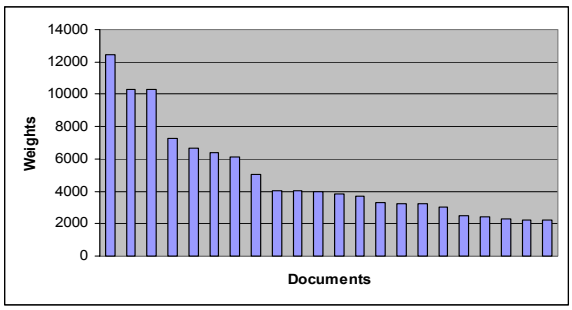

Figure 25. Weights of documents in the vegetarian cooking repository from document sets returned by Ask Jeeves using 2-clan-based paragraph connections.

All five search engines show a high degree of dispersion in their repositories if only 22 documents are viewed as compared to the high centralized results shown in Section 5.1. Now let us see what a high degree of dispersion mean to the repository. As stated in Section 4, the numbers of incoming links are almost the same for each document in the repository in case of a high degree of dispersion. There is no distinct kernel of documents identified as the most similar relevant and foraging activities may lead to a "lost in the hyperspace" feeling on the part of the user, who might lose track of his location within the overall hypertext space [20]. Obviously this is certainly not desirable as well as a high degree of centrality. Typically a user will have difficulties in finding an appropriate path if he is lost in the information navigation activity because he is now facing a situation where there are no distinct indications of the potential destination and every path leading to some others is possible to take. This means the augmented 2-clan P-Graph algorithm adds more than necessary links to the basic 5-gram method. This also has motivated our intentions of finding a more desirable P-Graph for its corresponding D-Graphs.

But if we increase the number of documents in the repositories to be tested, the results are different. Figure 26 and Figure 27 show the results of dispersion for both repositories when the number of documents in each is increased up to 77. A relationship of the number of documents in the repository and the degree of dispersion suggests that the D-Graphs resulting from P-Graphs augmented with extra links as described in the previous section have little trapping since almost all documents are accessible. Meanwhile, the centrality property is still present so that excessive dispersion is not a problem for the larger repositories. We also show the results of increasing the number of documents tested up to 40 in the repositories generated through the other four search engines as shown in Figure 28 through Figure 35.

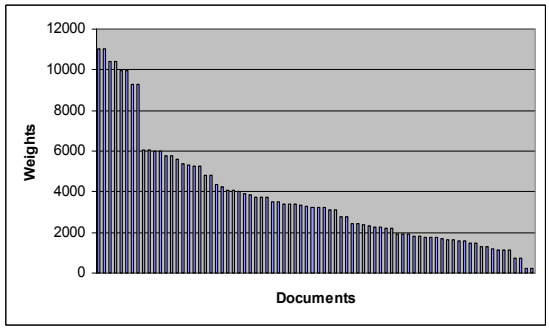

Figure 26. Weights of documents in the powered parachuting repository from document sets returned by AltaVista using 2-clan-based paragraph connections.

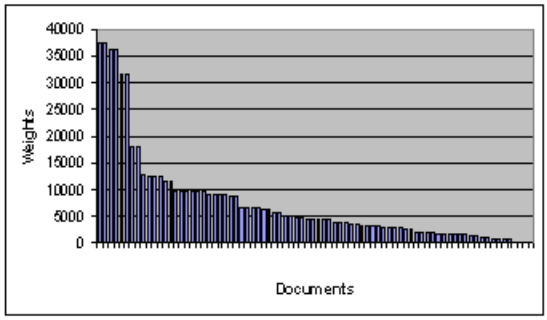

Figure 27. Weights of documents in the vegetarian cooking repository from document sets returned by AltaVista using 2-clan-based paragraph connections.

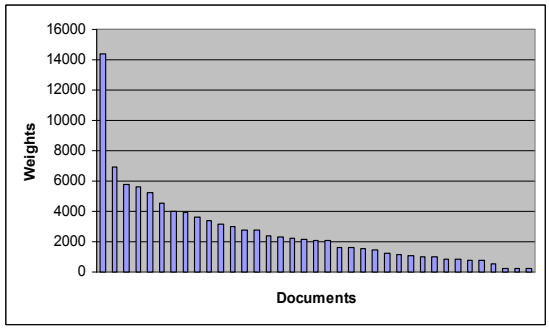

Figure 28. Weights of documents in the powered parachuting repository from document sets returned by Google using 2-clan-based paragraph connections.

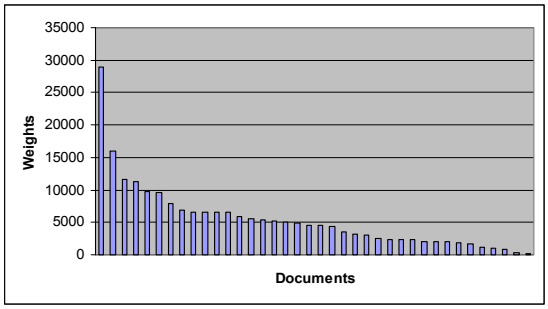

Figure 29. Weights of documents in the vegetarian cooking repository from document sets returned by Google using 2-clan-based paragraph connections only.

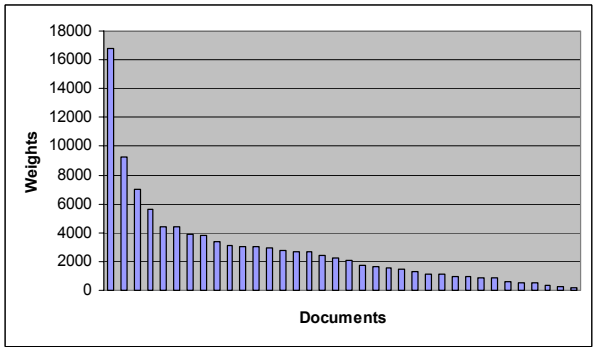

Figure 30. Weights of documents in the powered parachuting repository from document sets returned by Hotbot using 2-clan-based paragraph connections.

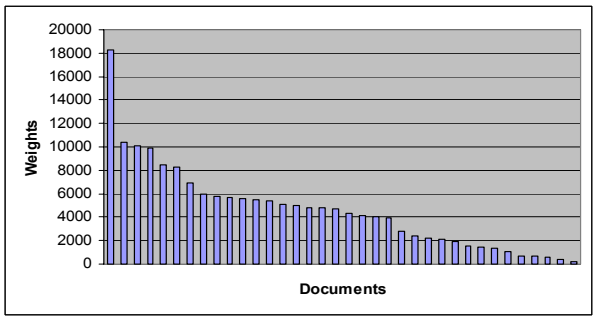

Figure 31. Weights of documents in the vegetarian cooking repository from document sets returned by Hotbot using 2-clan-based paragraph connections.

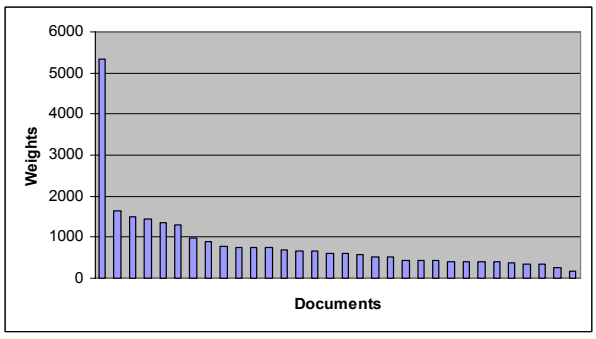

Figure 32. Weights of documents in the powered parachuting repository from document sets returned by Lycos using 2-clan-based paragraph connections.

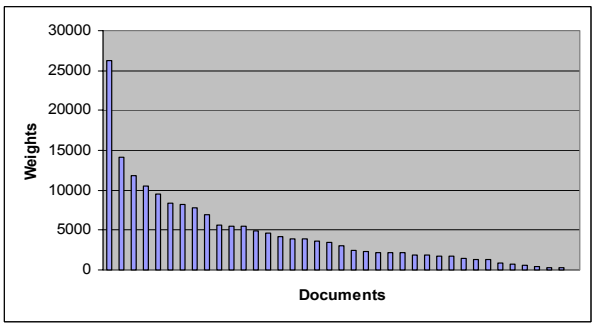

Figure 33. Weights of documents in the vegetarian cooking repository from document sets returned by Lycos using 2-clan-based paragraph connections.

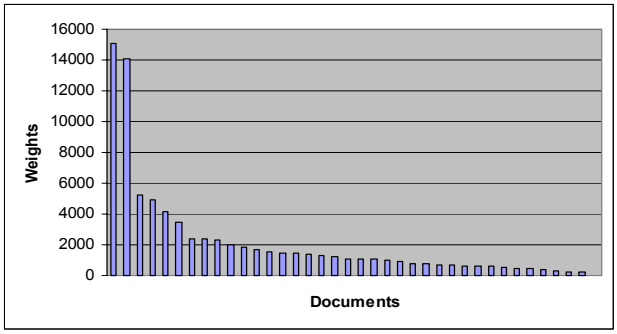

Figure 34. Weights of documents in the powered parachuting repository from document sets returned by Ask Jeeves using 2-clan-based paragraph connections.

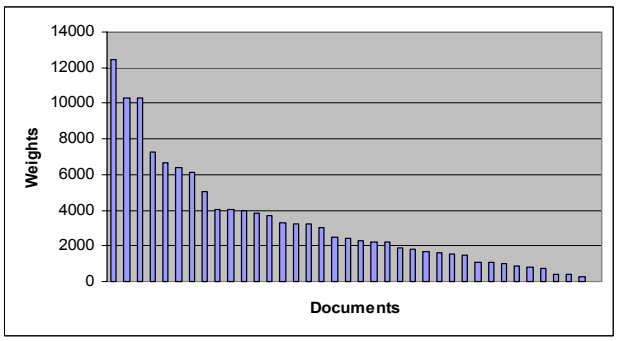

Figure 35. Weights of documents in the vegetarian cooking repository from document sets returned by Ask Jeeves using 2-clan-based paragraph connections.

6. Navigation with Ring-Structure in Centralized Repositories

We now have proposed a ring-structured solution to navigate in repositories with centrality but without a trapping problem, such as those with D-Graphs as shown in Figure 26 through Figure 35. The ring structure consists of a kernel meaning the documents with highest centralities, and two concentric rings around it meaning documents of intermediate and low weights respectively (shown in Figure 36). The ring closest to the kernel, called the near ring, contains documents of intermediate weights. The other ring, called the far ring, contains documents of low weights.

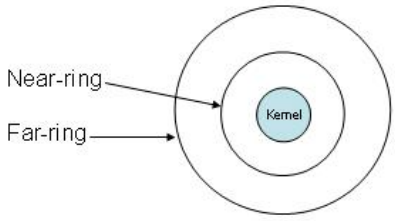

Figure 36. Ring-structured solution: kernel, near ring and far ring.

During the process of navigation, the user can see paragraph links that are used to represent whether they point to documents in the kernel, near ring or far ring. Therefore the user is able to decide which links to follow when foraging with the help of the ring-structured solution. The kernel would be intuitively associated with documents of high relevance to whatever criteria or query was used to obtain documents for the repository. The near ring would intuitively contain documents that are likely to be of interests, if, for example, foraging with just the kernel becomes boring. The far ring would contain documents peripherally related to the focus of the repository and thus likely be judged suitable for more expansive and undirected browsing. As stated in Section 4.1 and 5.1, each paragraph has 6 outgoing links. Let us k be the number of outgoing links pointing to paragraphs contained in kernel documents. Likewise, let n and f be the numbers of outgoing links pointing to paragraphs contained in the near and far ring documents respectively. Obviously we have the following equation:

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

We introduce a parameter r to approximately compute the quality of each paragraph by giving the following formula:

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

If n = 0 and f = 0 that means all outgoing links are pointing to paragraphs contained kernel documents, r = 100% indicating a good paragraph. If n = 6 and f = 0 that means all outgoing links are pointing to paragraphs contained in near ring documents, r is close to 50% indicating the quality of the paragraph is approximately half of good paragraphs. On the other side, if n = 0 and f = 6 that means all outgoing links are pointing to paragraphs contained in far ring documents, r is close to 0 indicating a bad paragraph. The numerical measurement of paragraphs helps fix the trapping problem for users by attaching a tuple (r, k, n, f) to each paragraph.

7. Conclusions

The D-Graphs constructed by five search engines show the tendency of high centrality and trapping problem when the P-Graph is built by 5-gram algorithms, although their Effectiveness of Construction (EOC) values vary. We therefore investigated ways of add facilitating links among paragraphs, and determined whether they imply more favorable D-Graphs. We found that D-Graphs based on 2-clan P-Graphs with the 3 types of links implied by Figure 5(b-d) let to more desirable D-Graphs: trapping avoidance, and moderate degrees of centrality and dispersion. This has been proven by repositories constructed through all the 5 search engines. Their EOC values are calculated and compared. Finally a ring-structured with numerical measurements was proposed to allow users to better control where they are in a repository without trapping problem, but with centrality.

References

[1] Yoshinori Hara, and Kojiro Watanabe. “Hypermedia Research at C&C Research Labs, NEC USA”, CHI’97 Electronic Publications: Organizational Overview, Mar. 22-27, 1997.

[2] Daniel Berleant, Jinghao Miao, and Zhong Gu. “Direct Multidisplay with MultiBrowser”, submitted.

[3] George W. Furnas. “Effective View Navigation”, CHI’97 Conference Proc., Mar. 22-27, 1997.

[4] Loren Terveen, and Will Hill. “Constructing, Organizing, and Visualizing Collections of Topically Related Web Resources”, ACM Trans. on Computer-Human Interaction, Vol. 6, No. 1, March 1999, pp 67-94.

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

[6] Eser Kandogan, and Ben Shneiderman,.“Elastic Windows: Evaluation of Multi-Window Operations”, CHI’97, Mar. 22-27, 1997.

[7] Sergey Brin and Lawrence Page. “The Anatomy of Large-Scale Hypertextual Web Search Engine”, In Proceedings of the WWW’7 Conferences, 1998.

[8] Peter Pirolli, and Stuart Card. “Information Foraging in Information Access Environments”, In the Proceedings of CHI’95, 1995.

[9] Frank G. Halasz. “Reflections on Notecards: Seven Issues For the Next Generation of Hypermedia Systems”, Communications of the ACM, 1988.

[10] Sara A. Bly, and Jarrett K. Rosenberg. “A Comparison of Tiled and Overlapping Windows”, Proceedings of CHI’86, 1986.

[11] Daniel Berleant and Hal Berghel. “Customizing Information: Part 1.” Computer 27 (9) (Sept. 1994) 96-98. “Part 2.” Computer 27 (10) (Oct. 1994) 76-78.

[12] Damashek. “Gauging Similarity With N-Grams: Language-Independent Categorization of Text.” Science 267 (10 Feb. 1995) 843-848.

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

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

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

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

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

[18] L. Tauscher and S. Greenberg. “Revisitation Patterns in World Wide Web Navigation.” Proceeding CHI’97, March, 399-406, ACM Press.

[19] T. Tse, G. Marchionini, W. Ding, L. Slaughter, and A. Komlodi. “Dynamic Key Frame Presentation Techniques for Augmenting Video Browsing.” Proceedings AVI’98: Advanced Visual Interfaces, 185-194.

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

[21] Jinghao Miao, and Dan Berleant. “Graph Structures in Paragraph-Linked Repositories”, submitted.